Christopher Doer

Multisensorielle Navigation und Kartierung in schwierigen Bedingungen

KIT Scientific Publishing

Christopher Doer

Multisensorielle Navigation und Kartierung in schwierigen Bedingungen

Multisensorielle Navigation und Kartierung in schwierigen Bedingungen

von
Christopher Doer

KIT Scientific Publishing

Karlsruher Institut für Technologie
Institut für Regelungs- und Steuerungssysteme (IRS)

Multisensorielle Navigation und Kartierung
in schwierigen Bedingungen

Zur Erlangung des akademischen Grades eines Doktor-Ingenieurs
von der KIT-Fakultät für Elektrotechnik und Informationstechnik des
Karlsruher Instituts für Technologie (KIT) genehmigte Dissertation

von Christopher Doer, M.Sc.

Tag der mündlichen Prüfung: 18. Juli 2022
Hauptreferent: Prof. Dr.-Ing. Gert F. Trommer
Korreferent: Prof. Dr.-Ing. Dieter Moormann

Impressum

KIT Scientific Publishing

Karlsruher Institut für Technologie (KIT)
KIT Scientific Publishing
Straße am Forum 2
D-76131 Karlsruhe

www.ksp.kit.edu

Print on Demand 2022 – Gedruckt auf FSC-zertifiziertem Papier

ISBN 978-3-7315-1240-0
DOI 10.5445/KSP/1000151323

Vorwort

Die vorliegende Arbeit entstand während meiner Tätigkeit als wissenschaftlicher Mitarbeiter am Institut für Theoretische Elektrotechnik und Systemoptimierung (ITE), seit 2020 Institut für Regelungs- und Steuerungssysteme (IRS) des Karlsruher Instituts für Technologie (KIT).

Mein besonderer Dank gilt Prof. Dr.-Ing. Gert F. Trommer für die Ermöglichung und Förderung dieser Arbeit sowie die vertrauensvolle Zusammenarbeit und wertvollen Anregungen. Auch möchte ich mich bei Prof. Dr.-Ing. Dieter Moormann für die Übernahme des Korreferats und das Interesse an meiner Arbeit bedanken.

Meinen Kolleginnen und Kollegen, die ich am ITE und IRS kennenlernen durfte, möchte ich für die gute Zusammenarbeit, die angenehme Arbeitsatmosphäre, die konstruktiven Diskussionen und die hilfreichen Anmerkungen zu meiner Arbeit danken. Weiterhin danke ich allen Studierenden, die mit ihren Bachelor- und Masterarbeiten zum Gelingen dieser Arbeit beigetragen haben.

Ganz herzlich möchte ich mich auch bei Katharina sowie meiner Familie für den Rückhalt und die Unterstützung bedanken.

Karlsruhe, im Juli 2022

Christopher Doer

Inhaltsverzeichnis

Abbildungsverzeichnis

Tabellenverzeichnis

Abkürzungen und Symbole

Abkürzungen

ATE	Absolute Translation Error
CEP	Circular Error Probable
CNN	Convolutional Neural Network
DRZ	Deutsches Rettungsrobotik-Zentrum
EB	Entscheidungsbaum
FAST	Features from Accelerated Segment Test
FFT	Fast Fourier Transform
FMCW	Frequency Modulated Continous Wave
GDOP	Geometric Dilution of Precision
GNSS	Global Navigation Satellite System
GPS	Global Positioning System
IMU	Inertiale Messeinheit
INS	Inertiales Navigationssystem
IoU	Intersection over Union
LR	Logistische Regression
LSQ	Least Squares

MEMS	Microelectromechanical Systems
MW	Manhattan Welt
MWO	Manhattan Welt Objekt
NED	North East Down
ODR	Orthogonal Distance Regression
RANSAC	Random Sample Consensus
RCS	Radar Cross Section
RGB	Red Green Blue
RGBD	Red Green Blue Depth
RIO	Radar inertiale Odometrie
RMS	Root Mean Square
ROS	Robot Operating System
RF	Random Forest
SLAM	Simultaneous Localization and Mapping
SPI	Serial Peripheral Interface
SVM	Support Vector Machine
UART	Universal Asynchronous Receiver / Transmitter
UAS	Unmanned Aerial System
USB	Universal Serial Bus
VIO	Visuelle inertiale Odometrie
VTOL	Vertical Takeoff and Landing Vehicle

Symbole und Variablen

Matrizen

$\mathbf{C}$	Richtungskosinusmatrix
$\mathbf{H}_H$	Homographiematrix
$\mathbf{H}_k$	Messmatrix
$\mathbf{I}$	Einheitsmatrix
$\mathbf{J}$	Jacobimatrix
$\mathbf{K}$	Intrinsische Kameramatrix
$\mathbf{K}_k$	Kalman-Gain-Matrix
$\mathbf{P}$	Kovarianzmatrix einer Messung
$\mathbf{P}_k$	Kovarianzmatrix des Systemzustands
$\mathbf{Q}_k$	Kovarianzmatrix des Systemrauschens $\boldsymbol{w}_k$
$\mathbf{R}_k$	Kovarianzmatrix des Messrauschens $\boldsymbol{v}_k$
$\mathbf{\Phi}_k$	Transitionsmatrix

Vektoren

$\boldsymbol{a}$	Beschleunigung
$\boldsymbol{b}_a$	Beschleunigungsmesseroffset
$\boldsymbol{b}_\omega$	Drehratensensoroffset
$\boldsymbol{d}$	Verbindungsvektor
$\boldsymbol{e}$	Fehler
$\boldsymbol{l}$	Translation
$\boldsymbol{\omega}$	Drehrate

$\boldsymbol{p}$	Position
$\boldsymbol{\Psi}$	Roll-Pitch-Gier-Eulerwinkel
$\boldsymbol{q}$	Quaternion
$\boldsymbol{u}_k$	Eingangsvektor
$\boldsymbol{v}$	Geschwindigkeit
$\boldsymbol{v}_k$	Messrauschen
$\boldsymbol{w}_k$	Systemrauschen
$\boldsymbol{x}_k$	Systemzustand
$\boldsymbol{y}_k$	Messwerte

Skalare

B	Bandbreite
b_b	Barometerbias
d	Distanz
g	Betrag der Erdbeschleunigung $\boldsymbol{g}$
λ	Wellenlänge
G	Antennengewinn
g	Betrag der Erdbeschleunigung
γ	Manhattan-Welt-Winkel
ϕ_{ab}	Rollwinkel der Rotation vom Koordinatensystem b nach a
ϕ_E	Elevationswinkel
ψ_{ab}	Gierwinkel der Rotation vom Koordinatensystem b nach a
θ_{ab}	Nickwinkel der Rotation vom Koordinatensystem b nach a
θ_A	Azimutwinkel

σ^2	Varianz
σ_{RCS}	Radarquerschnitt eines Objektes
T	Zeitdauer
t	Zeit

Operatoren und mathematische Symbole

$x, \boldsymbol{x}, \mathbf{X}$	Skalar, Vektor und Matrix
$\boldsymbol{f}(\boldsymbol{x}_k, \boldsymbol{u}_k)$	Systemfunktion
$\boldsymbol{h}(\boldsymbol{x}_k)$	Messfunktion
$p()$	Wahrscheinlichkeit von ()

Indizes und Sonderzeichen

$\hat{\square}$	Geschätzte Größe
$\tilde{\square}$	Fehler der Größe
$\bar{\square}$	Mittelwert
$\square^-$	A-priori Größe
$\square^+$	A-posteriori Größe
$\square^{\mathrm{x}}$	Größe angegeben im Koordinatensystem x
$\lvert\square\rvert$	Skalarer Betrag
$[\boldsymbol{v}]\times$	Schiefsymmetrische Matrix zum 3×1-Vektor $\boldsymbol{v}$
$\otimes$	Quaternionenmultiplikation nach Hamilton Definition
$\circ$	Skalarprodukt

1 Einleitung

Unbemannte elektrische Fluggeräte verschiedenster Bauart und Größe finden immer mehr Anwendung in vielfältigen Einsatzbereichen. Laut aktueller Studien [87, 149] werden derartige Drohnen bereits heute im professionellen Bereich erfolgreich bei der Vermessung, Inspektion, Filmaufnahmen und Kartierung eingesetzt. Darüber hinaus erfährt der Transport eiliger Güter, wie Medizin [119,132] oder Lieferungen [52,104], große Aufmerksamkeit. Zahlreiche Pilotprojekte, wie beispielsweise von Wing [155], Amazon Prime Air [7] oder Wingcopter [153] demonstrieren bereits das große Potenzial. Auch bei Such- und Rettungsmissionen kann durch den Einsatz von kleinen unbemannten Fluggeräten ein erheblicher Mehrwert generiert werden [65, 104]. Dies wurde in verschiedenen Forschungsprojekten, wie zum Beispiel VISION [146] oder FALKE [42], erfolgreich demonstriert.

Aktuelle Fluggeräte benötigen jedoch in der Regel einen Piloten zur Steuerung und Überwachung. Bei unbemannten Systemen erzeugt der erforderliche Pilot stark erhöhte Kosten, was die Skalierbarkeit zum Beispiel bei der Lieferung durch Drohnen verhindert. Außerdem schränkt der Betrieb innerhalb der Sichtweite des Piloten mögliche Anwendungen sehr stark ein. So erfordern Such-, Rettungs- und Kartierungsmissionen meist Flüge außerhalb der Sichtweite.

Der Schlüssel zur weiteren Verbreitung von Fluggeräten in den genannten Anwendungen ist folglich die Autonomie dieser Systeme. Dies ist daher Gegenstand aktueller Forschung. Der Fokus liegt dabei sowohl auf der Entwicklung von robusten Navigationssystemen und Verfahren zur Kollisionsvermeidung als auch der Integration in den Luftraum. Um die herausfordernden Randbedingungen der genannten Praxisanwendungen zu lösen, sind verschiedene Sensoren für das Fluggerät notwendig, die sich komplementär ergänzen. Die stetige Weiterentwicklung und Verbesserung leistungsfähiger Sensorik bietet das große Potenzial robuste autonome Flugsysteme zu entwickeln.

1.1 Motivation

Für autonome Fluggeräte ist eine sehr genaue Kenntnis der Navigationslösung, also der Position, Geschwindigkeit und Lage, entscheidend, um eine präzise Flugführung und damit einen sicheren Flug zu gewährleisten. Basierend auf der Navigationslösung kann eine Kartierung der Umgebung durchgeführt werden. Dies stellt beispielsweise bei Such- und Rettungsmissionen das Lagebild zur weiteren Einsatzplanung bereit. Zusätzlich wird dadurch die Detektion von Hindernissen ermöglicht, was die Grundlage für die Planung kollisionsfreier Trajektorien ist.

Navigation und Kartierung werden bereits seit Jahren intensiv erforscht und Systeme mit hoher Autonomie wurden entwickelt. So sind autonome Quadrokopter-Flüge mit Hilfe von visuellen Sensoren wie Laserscannern und Kameras demonstriert worden [89, 110, 128, 134]. Auch kommerzielle Fluggeräte, wie zum Beispiel von der Firma DJI [35] oder Skydio [136], verfügen über Kamera-basierte Systeme zur Kollisionsvermeidung. Trotz der sehr guten Ergebnisse der Kamera- und Laser-basierten Ansätze können diese nur bei guten Sichtbedingungen eingesetzt werden. Anwendungen wie Lieferdienste oder auch Such- und Rettungsmissionen sollen aber auch bei schlechten Sichtbedingungen verfügbar sein. Somit muss das Navigationssystem trotz Sichteinschränkungen durch Nebel, Rauch, Dunkelheit oder Blendung einen sicheren Flug gewährleisten und eine Kartierung der Umgebung ermöglichen. Weiterhin sind autonome Missionen von kommerziellen Drohnen mit Hilfe des Globalen Navigationssatellitensystems (GNSS) möglich und werden bereits beispielsweise bei Kartierungs- und Suchaufgaben eingesetzt [154]. Aufgrund von Mehrwegeausbreitung oder Abschattung von Gebäuden in urbanen Gebieten kann die GNSS-Genauigkeit und Verfügbarkeit stark eingeschränkt werden. Auch Bedrohungen durch Störung (Jamming) oder Verfälschung (Spoofing) der GNSS-Signale kann den sicheren Flugbetrieb erheblich einschränken.

In der vorliegenden Dissertation werden daher Ansätze erarbeitet, die eine sichere Navigation und umfassende Kartierung der Umgebung auch in diesen herausfordernden Bedingungen ermöglichen. Dabei werden neuartige multisensorielle Konzepte entwickelt, umfangreich evaluiert und deren hohe Leistungsfähigkeit in der Praxisanwendung autonomer Flüge auch bei Einschränkung der Sicht und des GNSS-Empfangs demonstriert.

1.2 Szenario

In dieser Arbeit wird das in Abbildung 1.1 dargestellte Szenario betrachtet. Nach dem Start wird der in grün dargestellte Flugpfad abgeflogen und anschließend wieder gelandet. Der Flug besteht dabei aus drei Phasen: Start, Mission und Landung. All diese Phasen sollen komplett autonom ausgeführt werden. Während der Mission kann der Flug sowohl durch urbane Umgebungen als auch durch Gebäude führen. Dabei soll die Umgebung einschließlich Personen kartiert werden. Es können in allen drei Phasen herausfordernde Bedingungen auftreten, die den Betrieb der bisher am Markt erhältlichen Systeme verhindern würde:

- Einschränkung der Sichtbedingungen durch Rauch, Nebel, Dunkelheit oder Blendung.
- Stark schwankende GNSS-Bedingungen sowie der Ausfall von GNSS auf Grund von Abschattung oder durch Bedrohungen wie Jamming oder Spoofing.

Vielfältige Anwendungsfelder können im Rahmen dieses Szenarios betrachtet werden. So müssen bei allen Fluganwendungen sichere Start- und Landemanöver gewährleistet sein. Um eine hohe Verfügbarkeit der Systeme zu

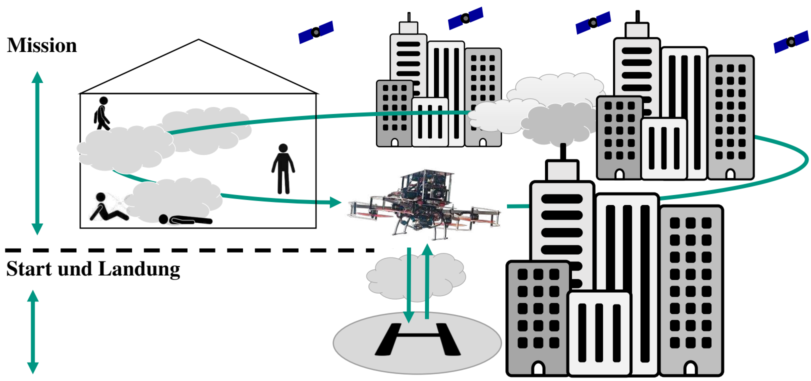


Abbildung 1.1: Im Rahmen dieser Dissertation betrachtetes Szenario

ermöglichen, muss dies auch bei schwierigen Bedingungen sichergestellt sein. Für die Navigation in urbanen Gebieten wird ein globales Navigationssystem benötigt, was beispielsweise auch für die Urban Air Mobility, Lieferdienste, Wartung oder Überwachung eingesetzt werden kann. Eine Kartierung der Umgebung kann für die globale Bahnplanung und Kollisionsvermeidung genutzt werden. Der Flug in der Nähe oder sogar innerhalb eines Gebäudes trotz Sichteinschränkungen durch Rauch oder Nebel, einschließlich der Kartierung von Personen, bildet die Anwendung von Such- und Rettungsmissionen ab. Zur Rettung ist der Zustand der Personen elementar, somit soll die erstellte Karte Informationen hierzu enthalten. Das betrachtete Szenario deckt also vielfältige und komplexe Anwendungsfelder ab. Schwerpunkt ist dabei die Navigation des Fluggerätes sowie die Kartierung der Umgebung.

1.3 Ziele

Ziel dieser Arbeit ist es, Verfahren zur Navigation eines Fluggeräts sowie der Kartierung der Umgebung im zuvor beschriebenen Szenario zu entwickeln, welche die Autonomie des Flugsystems ermöglichen. Daraus ergeben sich folgende Ziele für die vorliegende Dissertation:

- Die Auswahl und Entwicklung eines Sensorkonzeptes, das den autonomen Betrieb im motivierten Szenario ermöglicht.
- Der Entwurf und die Implementierung von Verfahren für autonome Start- und Landemanöver, die einen sicheren Betrieb auch bei schwierigen Bedingungen ermöglichen.
- Die Entwicklung von Verfahren zur sicheren Navigation von Fluggeräten auch in schwierigen Sichtbedingungen beispielsweise durch Nebel, Rauch, Dunkelheit oder Blendung.
- Die Entwicklung eines Navigationssystems, das eine global referenzierte Navigationslösung sowohl in schlechten Sichtbedingungen als auch schwierigen GNSS-Bedingungen durch verfälschte GNSS-Signale, beispielsweise verursacht durch Abschattung, Spoofing oder Jamming, liefert.

- Die Kartierung der Umgebung in Form einer Rasterkarte sowie die Kartierung von Personen innerhalb dieser Karte selbst bei schwierigen Sichtbedingungen, wobei die Körperhaltung der Personen hinsichtlich *aufrecht* stehend oder laufend, *sitzend* oder *liegend* klassifiziert wird.
- Die Navigation und Kartierung soll auch für urbane Gebiete außerhalb von Gebäuden sowie in Innenräumen für Flüge eingesetzt werden können.
- Die Prozessierung soll auf dem Embedded-Computer des Fluggeräts und ohne Verbindung zu einer Bodenstation in Echtzeit ausgeführt werden.
- Die entwickelten Verfahren sollen umfassend mit Realdatensätzen evaluiert und gegenüber etablierten Verfahren unter guten Sichtbedingungen verglichen werden.
- Die entwickelten Navigations- und Kartierungsverfahren sollen unter realen Bedingungen in schlechten Sicht- und GNSS-Bedingungen für autonome Flüge demonstriert werden und dabei eine Karte in Echtzeit erstellen, die als Lagebild für einen Rettungseinsatz eingesetzt werden kann.

1.4 Aufbau der Arbeit

Die vorliegende Arbeit ist in acht Kapitel eingeteilt. Im nachfolgenden Kapitel 2 werden zunächst relevante Grundlagen eingeführt. Es wird die Nomenklatur definiert und auf wichtige Aspekte der Navigation, der Radar-Technik sowie der Personendetektion mit neuronalen Netzen eingegangen.

Der entwickelte Hardwareaufbau wird in Kapitel 3 vorgestellt. Hierbei wird zunächst die Auswahl der Sensorik für das motivierte Szenario aus Abbildung 1.1 dargelegt. Anschließend wird die entworfene Multisensorplattform sowie die Flugplattform detailliert beschrieben.

Der jeweilige Stand der Technik wird in jedem Kapitel des Hauptteils (Kapitel 4 - 7) jeweils analysiert und diese Kapitel werden mit umfangreichen experimentellen Ergebnissen abgeschlossen.

Im Kapitel 4 werden zwei Verfahren für autonome Start- und Landemanöver entwickelt. Beide Verfahren ermöglichen die Start- und Landemanöver unter schwierigen Sichtbedingungen. Umfangreiche Testflüge demonstrieren beide Verfahren und ein abschließender Vergleich zeigt die Vor- und Nachteile auf.

Im Kapitel 5 wird ein neuartiges System zur Radar-inertialen Navigation beschrieben. Dabei wird zunächst auf die Radar-Eigengeschwindigkeitsschätzung sowie die Radar-Gierwinkelstützung eingegangen und beide Stützinformationen evaluiert. Darauf folgt die Integration der Radar-Stützinformationen mit Inertialsensordaten im Radar-inertialen Navigationsfilter. Evaluation in Simulation sowie umfangreiche Realdatensätze und die Demonstration autonomer Flüge werden präsentiert.

Aufbauend auf Kapitel 5 erfolgt im Kapitel 6 zunächst die Erweiterung eines etablierten Kamera-basierten Verfahrens mit Radar-Eigengeschwindigkeitsmessungen. Anschließend folgt die Weiterentwicklung der Radar-inertialen Navigation zu einem globalen Navigationssystem, was bei schwierigen Sichtbedingungen aber auch bei herausfordernden GNSS-Bedingungen eingesetzt werden kann.

Schließlich erfolgt in Kapitel 7 die Vorstellung der multisensoriellen Kartierung. Basierend auf der Radar-inertialen Navigation wird ein Radar-SLAM (Simultaneous Localization and Mapping) entworfen, der die Umgebung mit einer Rasterkarte modelliert. Anschließend erfolgt die Entwicklung von Ansätzen zur Detektion von Personen sowie die Klassifizierung der Körperhaltung basierend auf neuronalen Netzen. Die Position der detektierten Personen wird schließlich in der Rasterkarte verankert. Dies führt zu einem umfassenden Lagebild, dass eine gezielte Rettung der Personen ermöglicht.

Die Zusammenfassung der Arbeit erfolgt abschließend in Kapitel 8.

2 Grundlagen

In diesem Kapitel wird zunächst in die eingesetzte Nomenklatur eingeführt und verwendete Koordinatensystem definiert. Anschließend erfolgt die Erläuterung wesentlicher Grundlagen der Navigation, der Radartechnik sowie des maschinellen Lernens. Abschließend werden die eingesetzten Metriken zur Evaluation von Navigationsfiltern sowie von Personendetektion und -klassifizierung zusammengefasst.

2.1 Nomenklatur und Koordinatensysteme

Nomenklatur

Die eingesetzte Nomenklatur orientiert sich am Standartwerk von Groves [53]. In dieser Arbeit werden **skalare Größen** mit *normal* gedruckten Buchstaben gekennzeichnet (z.B. n oder T). **Vektoren** werden *kleingeschrieben* mit *fettgedruckten Buchstaben* (z.B. $\boldsymbol{a}^b$). Dabei gibt der obere Index das Koordinatensystem an, in welchem der Vektor definiert ist. Bei einem **Geschwindigkeitsvektor** $\boldsymbol{v}_{bc}^{a}$ ist die Geschwindigkeit des Koordinatensystems $\{\}^c$ bezüglich $\{\}^b$ im Koordinatensystem $\{\}^a$ angegeben. Die **Translation** $\boldsymbol{l}_{bc}^{a}$ beschreibt die Verschiebung des Koordinatensystems $\{\}^b$ bezüglich $\{\}^c$ im Koordinatensystem $\{\}^a$. **Matrizen** werden durch *großgeschriebene*, *fette Buchstaben* (z.b. **M**) gekennzeichnet. **Rotationen** werden mittels Eulerwinkeln, Rotationsmatrizen oder Quaternionen modelliert werden. Die drei Eulerwinkel bestehen aus dem **Rollwinkel** θ_{ab}, dem **Pitchwinkel** ϕ_{ab} und dem **Gierwinkel** ψ_{ab} und beschreiben dabei die Rotation von $\{\}^b$ nach $\{\}^a$ mit Rotationsreihenfolge zyx. Dies wird auch durch die **Rotationsmatrix** $\mathbf{C}_b^a$ oder das **Quaternion** $\boldsymbol{q}_b^a$ ausgedrückt. Die Quaternionen folgen der Hamilton-Definition, für eine detaillierte Definition sei auf [137] verwiesen.

Koordinatensysteme

Ein Koordinatensystem kann durch den Ursprung sowie der zugehörigen Achsen definiert werden. In dieser Arbeit werden ausschließlich rechtshändige, orthogonale Koordinatensysteme eingesetzt. Dabei werden die folgenden Koordinatensysteme, die sich an [53] orientieren, unterschieden:

- Das **erdfeste-Koordinatensystem** (*earth-frame*, $\{\}^e$) hat seinen Ursprung im Mittelpunkt des Rotationsellipsoids der Erde, wobei die Achsen fest in Bezug auf die Erde sind. Die x^e- und y^e-Achsen sind parallel zur Äquatorialebene und die z^e-Achse zeigt in Richtung des geographischen Nordpols. Es wird häufig auch Earth-centered, Earth-fixed (ECEF)-Koordinatensystem genannt [152].
- Das **körperfeste-Koordinatensystem** (*body-frame*, $\{\}^b$) ist fest in Bezug zum Körper der mobilen Plattform. Dabei zeigt die x^b-Achse vorwärts, die y^b-Achse nach rechts und die z^b-Achse nach unten. In dieser Arbeit wird der Ursprung des körperfesten-Koordinatensystems durch die Inertiale Messeinheit (Inertial Measurement Unit - IMU) definiert.
- Das **Navigationskoordinatensystem** (*navigation-frame*, $\{\}^n$) hat den gleichen Ursprung wie das körperfeste-Koordinatensystem. Die x^n-Achse zeigt dabei immer nach Norden, die y^n-Achse nach Osten und die z^n-Achse nach unten was der *north-east-down* (NED) Konvention folgt.
- Das **lokale-Koordinatensystem** (*origin-frame*, $\{\}^o$) kann seinen Ursprung an einem beliebigen festen Punkt auf der Erde haben, wobei die z^o-Achse analog zum Navigationskoordinatensystem nach unten zeigt. Die x^o- und y^o-Achsen spannen die horizontale Ebene auf.
- Das **Karten-Koordinatensystem** (*map-frame*, $\{\}^m$) hat einen festen Ursprung gegenüber dem lokalen-Koordinatensystem. Die z^m-Achse zeigt nach oben und die x^m- und y^m-Achsen liegen in der horizontalen Ebene.
- Das **stabilisierte-Koordinatensystem** (*stabilized-frame*, $\{\}^s$) hat den gleichen Ursprung wie das körperfeste-Koordinatensystem. Die z^s-Achse fällt mit der z^n-Achse zusammen und die x^s- und y^s-Achsen sind so ausgerichtet, dass die x^s-Achse vorwärts zeigt. Die Transformation aus dem körperfesten- in das stabilisierte-Koordinatensystem erfolgt mittels des Roll- und Pitchwinkels der Plattform: $\mathbf{C}_b^s(\theta_{sb}, \phi_{sb}, 0)$.

- Das **Radar-Koordinatensystem** (*radar-frame*, $\{\}^r$) hat seinen Ursprung im Zentrum der Radarantenne und weist die gleichen Achsenausrichtung wie das körperfeste-Koordinatensystem auf.
- Das **Kamera-Koordinatensystem** (*camera-frame*, $\{\}^c$) hat seinen Ursprung im Projektionszentrum der Kamera. Die z^c-Achse zeigt in Blickrichtung der Kamera, die x^c-Achse nach rechts und die y^c-Achse nach unten.

2.2 Grundlagen der integrierten Navigation

Integrierte Navigation ist durch die Kombination von verschiedenen Sensoren sowie Navigationsverfahren gekennzeichnet [152]. Dabei zielt die integrierte Navigation darauf ab, dass die Vorteile des einen Sensors oder Verfahrens die Nachteile eines anderen ausgleichen. Kernbestandteil von integrierten Navigationssystemen ist die inertiale Navigation. Diese kann eine Navigationslösung, also die Position, Geschwindigkeit und Lage auf Basis der Messdaten einer IMU berechnen. Um akkumulierende Fehler der Navigationslösung zu korrigieren, nutzen integrierte Navigationssysteme weitere Stützinformation in Form von Messungen und fusionieren diese beispielsweise mit einem Kalman-Filter [63]. Um auch verzögerte Messungen zu fusionieren können Kalman-Filter mit Zustandserweiterung durch Stochastic-Cloning [121] erweitert werden. Auf die inertiale Navigation und Kalman-Filter wird im Folgenden näher eingegangen.

2.2.1 Inertiale Navigation

Ein Inertialnavigationssystem (INS) berechnet aus den Daten einer IMU, also der Beschleunigung und Drehrate sowie einem Anfangszustand, die Navigationslösung. Die Messung der Beschleunigung und Drehrate erfolgt dabei für alle drei orthogonal zueinander ausgerichteten Achsen. Hierbei wird die IMU in modernen INS fest auf der mobilen Plattform verbaut, was auch als Strapdown-Konfiguration bezeichnet wird [53]. Die Berechnung der Navigationslösung aus den Messdaten einer IMU wird Strapdown-Berechung genannt [152]. Zunächst erfolgt durch Integration der gemessenen Drehraten die Propagation

der Lage. Basierend darauf kann die Geschwindigkeit mit Hilfe der um die Erdbeschleunigung bereinigten Beschleunigungsmessung propagiert werden. Die Integration der Geschwindigkeit liefert schließlich die Position. Für eine detaillierte Beschreibung der Strapdown-Berechnung sei auf [53, 152] verwiesen. INS ist ein sogenanntes kurzzeitgenaues Navigationsverfahren, da ohne Korrekturen die Navigationsfehler mit der Zeit anwachsen. Ursachen dafür sind beispielsweise die endliche Abtastrate der Sensoren und insbesondere bei den eingesetzten MEMS-IMUs ein sich ändernder Offset der Beschleunigungs- und Drehratensensoren [152]. Eine Korrektur dieser Offsets ist daher elementar für eine genaue, inertial-gestützte Navigation. Dies kann mit integrierten Navigationssystemen gelöst werden, wobei die Fusion mit weiterer Sensorik häufig mit einem Kalman-Filter [63], wie im folgenden Abschnitt beschrieben, durchgeführt wird.

2.2.2 Kalman-Filter

Der Kalman-Filter wurde ursprünglich durch R. E. Kálmán vorgeschlagen [63]. Es handelt sich dabei um einen linearen Schätzfilter, der den Systemzustand in Form des Zustandsvektors $\boldsymbol{x}_k$ mit assoziierter Kovarianzmatrix $\mathbf{P}_k$ schätzt. Dabei wird der Systemzustand mit Hilfe des Systemmodells in den nächsten Zeitschritt propagiert. Im Filterupdate wird dann der Systemzustand mittels Stützinformation weiterer Sensoren oder Verfahren korrigiert. Durch die Modellierung mit Zustandsvektor und Kovarianzmatrix ist dabei eine optimale Fusion gegeben, falls es sich beim System- und Messrauschen um mittelwertfreies, weißes Gaußsches Rauschen handelt [53].

Error-State-Kalman-Filter

Bei Realsystemen sind sowohl das Systemmodell als auch die Messgleichung sehr häufig nichtlinear. In diesem Fall bietet sich der Einsatz eines Error-State-Kalman-Filters an. Dieser führt eine Linearisierung des System- und Messmodells durch. Zusätzlich erfolgt eine Aufteilung in einen Nominalzustand $\hat{\boldsymbol{x}}_k$ und Fehlerzustand $\tilde{\boldsymbol{x}}_k$. Im Error-State-Kalman-Filter wird dabei nur der

Fehlerzustand geschätzt und der Nominalzustand damit korrigiert. Der wahre Zustand $\boldsymbol{x}_k$ ist durch die Fehlerdefinition

$$\boldsymbol{x}_k = \hat{\boldsymbol{x}}_k - \tilde{\boldsymbol{x}}_k \tag{2.1}$$

gegeben. Die Propagation des A-posteriori Nominalzustands $\boldsymbol{x}_k^+$ erfolgt mittels des zeitdiskreten, nichtlinearen Systemmodells $\boldsymbol{f}(\boldsymbol{x}_k^+, \boldsymbol{u}_k)$ mit

$$\boldsymbol{x}_{k+1}^- = \boldsymbol{f}(\boldsymbol{x}_k^+, \boldsymbol{u}_k) + \mathbf{G}_k \boldsymbol{w}_k \tag{2.2}$$

und liefert den A-priori Nominalzustand $\boldsymbol{x}_{k+1}^-$. Der A-posteriori Fehlerzustand $\tilde{\boldsymbol{x}}_k^+$ wird nicht propagiert, da eine Korrektur des Nominalzustands nach jedem Filterupdate vorgenommen wird und dieser daher gleich Null ist. Durch Linearisierung um den aktuellen Nominalzustand $\hat{\boldsymbol{x}}_k$ ergibt sich die Transitionsmatrix zu

$$\mathbf{\Phi}_k = \left. \frac{\partial \boldsymbol{f}\left(\boldsymbol{x}_k, \boldsymbol{u}_k\right)}{\partial \boldsymbol{x}_k} \right|_{\boldsymbol{x}_k = \hat{\boldsymbol{x}}_k} \tag{2.3}$$

womit die Propagation der Kovarianz durch

$$\mathbf{P}_{k+1}^- = \mathbf{\Phi}_k \mathbf{P}_k^+ \mathbf{\Phi}_k^T + \mathbf{G}_k \mathbf{Q}_k \mathbf{G}_k^T \tag{2.4}$$

erfolgt. Die nichtlineare Messgleichung des Fehlerzustands wird durch

$$\tilde{\boldsymbol{y}}_k = \boldsymbol{h}\left(\hat{\boldsymbol{x}}_k^-\right) + \boldsymbol{v}_k \tag{2.5}$$

beschrieben. Linearisierung um den aktuellen Nominalzustand ergibt die Jacobimatrix:

$$\mathbf{H}_k \approx \left. \frac{\partial \boldsymbol{h}\left(\boldsymbol{x}_k\right)}{\partial \tilde{\boldsymbol{x}}_k} \right|_{\boldsymbol{x}_k = \hat{\boldsymbol{x}}_k} . \tag{2.6}$$

Basierend darauf kann die Kalman-Gain-Matrix

$$\mathbf{K}_k = \mathbf{P}_k^- \mathbf{H}_k^T \left(\mathbf{H}_k \mathbf{P}_k^- \mathbf{H}_k^T + \mathbf{R}_k\right)^{-1} \tag{2.7}$$

aufgestellt und das Filterupdate implementiert werden, was den A-posteriori Fehlerzustand $\tilde{\boldsymbol{x}}_k^+$ sowie die aktualisierte Kovarianzmatrix $\mathbf{P}_k^+$ liefert:

$$\tilde{\boldsymbol{x}}_k^+ = \mathbf{K}_k \left(\boldsymbol{h}\left(\hat{\boldsymbol{x}}_k^-\right) - \boldsymbol{y}_k \right) = \mathbf{K}_k \boldsymbol{r}_k \tag{2.8}$$

$$\mathbf{P}_k^+ = \mathbf{P}_k^- - \mathbf{K}_k \mathbf{H}_k \mathbf{P}_k^-. \tag{2.9}$$

Der A-posteriori Nominalzustand wird mit Hilfe der Fehlerdefinition aus (2.1) für Skalare und Vektoren mit:

$$\hat{\boldsymbol{x}}_k^+ = \hat{\boldsymbol{x}}_k^- - \tilde{\boldsymbol{x}}_k^+ \tag{2.10}$$

korrigiert und für die Lage folgt mit dem A-posteriori Fehlerzustand des Eulerwinkels $\tilde{\boldsymbol{\theta}}_k^+$

$$\hat{\boldsymbol{q}}_k^+ = \tilde{\boldsymbol{q}}_k^+ \otimes \hat{\boldsymbol{q}}_k^-, \qquad \tilde{\boldsymbol{q}}_k^+ \approx \begin{bmatrix} 1 & -1/2\, \tilde{\boldsymbol{\theta}}_k^{+,T} \end{bmatrix}^T \tag{2.11}$$

mit der Quaternionen-Multiplikation $\otimes$ [137]. Nach der Korrektur wird der Fehlerzustand zu Null gesetzt, daher wird dies auch Closed-Loop-Implementierung genannt [53].

Zustandserweiterung mit Stochastic-Cloning

Sensoren weisen typischerweise eine Latenz zwischen der Gültigkeit und der Verfügbarkeit der Messung zur Verarbeitung im Filterupdate auf. Eine naive Lösung für dieses Problem wäre es, das Navigationsfilter mit einer ausreichend großen Latenz, also zeitversetzt zu betreiben. Diese Arbeit zielt jedoch auf ein Echtzeitnavigationssystem ab. Daher wird zur Verarbeitung verzögerter Messwerte eine Zustandserweiterung, auch Stochastic-Cloning genannt [121], angewendet. Zum Zeitpunkt der Gültigkeit k_0 der Messung wird der Zustand erweitert:

$$\boldsymbol{x}_{k_0,e} = \begin{bmatrix} \boldsymbol{x}_{k_0} \\ \boldsymbol{x}_{k_0} \end{bmatrix}, \qquad \mathbf{P}_{k_0,e} = \begin{bmatrix} \mathbf{P}_{k_0} & \mathbf{P}_{k_0} \\ \mathbf{P}_{k_0} & \mathbf{P}_{k_0} \end{bmatrix}. \tag{2.12}$$

Bei der Propagation werden dann die erweiterten Zustände nicht verändert. Ist die Messung verfügbar, erfolgt das Filterupdate mit den erweiterten Zuständen und korrigiert damit auch den aktuellen Systemzustand.

2.3 Grundlagen der Radartechnik

Aufgrund der Robustheit gegenüber Umwelt- und Witterungsbedingungen wird Radar bereits in vielfältigen Anwendungen eingesetzt [91]. Dazu zählen beispielsweise die Luftraumüberwachung, die Umfeldwahrnehmung und Kollisionsvermeidung bei Flugzeugen aber auch zur Kartierung mittels Satelliten oder in Fahrerassistenzsystemen werden Radarsysteme genutzt. Besondere Aufmerksamkeit haben in den letzten Jahren Radarsensoren mit Wellenlängen im Millimeterbereich, was Frequenzen von 30-300 GHz entspricht, erfahren. Diese benötigen nur sehr kleine Antennen, wodurch sogar Radarchips mit integrierten Antennen [166] hergestellt werden können. Radar basiert auf Laufzeitmessung von Radiowellen, so wird die Distanz d zum Objekt durch

$$d = \frac{c\,T_s}{2} \tag{2.13}$$

mit der Lichtgeschwindigkeit c und der Signallaufzeit T_s bei einem Radarsystem mit identischer Sende-und Empfangsantenne beschrieben. Die Radargleichung dient zur Berechnung der Empfangsleistung und veranschaulicht die Messcharakteristik. Die Empfangsleistung ergibt sich zu [91]

$$P_e = P_s \frac{G_s\,G_e\,\lambda^2\,\sigma_{\mathrm{RCS}}}{(4\pi)^3\,r^4} \tag{2.14}$$

mit der Sendeleistung P_s, dem Gewinn der Sendeantenne G_s und Empfangsantenne G_e, der Wellenlänge λ, der Distanz r zwischen Objekt und Antenne unter der Annahme, dass die Sende-und Empfangsantenne nahe beieinander liegen und dem mittleren Radarquerschnitt (Radar Cross Section - RCS) σ_{RCS} des Objektes. Wie in (2.14) ersichtlich, geht die Distanz biquadratisch ein und P_e fällt daher mit steigender Entfernung sehr stark ab. Zusätzlich kommen bei Realsystemen noch weitere Verluste beispielsweise durch die Signalverarbeitung und atmosphärischer Dämpfung hinzu [91]. Traditionelle Radarsysteme verwenden die Pulsradar-Methode. Dabei wird ein kurzer Puls mit großer Leistung gesendet und über die Laufzeit mit (2.13) die Distanz bestimmt. Nachteilig ist dabei jedoch, dass diese Systeme eine Mindestdistanz benötigen, die durch die Pulsbreite limitiert wird. Zusätzlich sind sehr große Sendeleistung notwendig. Daher hat sich Frequency-Modulated-Continous-Wave (FMCW) Radar zur dominanten Radartechnik bei vielen Anwendungen entwickelt [61].

2.3.1 FMCW-Radar-Prinzip

Im Gegensatz zum Pulsradar, wird beim FMCW-Radar eine deutlich geringere Sendeleistung über einen deutlich längeren Zeitraum übertragen. Zusätzlich können auch sehr kurze Distanzen, bis in den niedrigen Zentimeterbereich, gemessen werden. Dabei wird ein Signal, auch Chirp genannt, mit ändernder Frequenz gesendet. Je nach Distanz zum Objekt ergibt sich so eine Frequenzdifferenz zwischen dem aktuellen Sende- und Empfangssignal. Die Mischung beider Signale mit anschließender Tiefpassfilterung ergibt das Differenzsignal, was deutlich niederfrequenter als die Sendefrequenz ist. Es kann daher mit Analog-Digital (AD)-Wandlern digitalisiert werden. Es wird hierbei meist ein Sägezahn-Signal eingesetzt, was einen linearen Frequenzhub aufweist. Problematisch dabei ist, dass eine mögliche Dopplerverschiebung, auf Grund von Bewegung des Objekts, auch die Differenzfrequenz beeinflusst. Abhilfe kann deshalb ein Dreieck-Signal schaffen, da hiermit die Frequenzänderung durch die Distanz und Geschwindigkeit voneinander getrennt werden können. Traditionelle FMCW-Radarsensoren senden immer nur einen einzelnen Chirp mit Zeitdauern im Bereich 1 – 10 ms. Dies benötigt nur sehr günstige AD-Wandler mit Samplingraten unterhalb 1 MHz [117].

2.3.2 Fast-Chirp-FMCW-Radar

Modernen FMCW-Radarsensoren, wie sie auch für Fahrerassistenzsysteme eingesetzt werden, nutzen ein modifiziertes Verfahren [117]. Diese sogenannten Fast-Chirp-FCMW-Radarsensoren übertragen eine Vielzahl an sehr kurzen Sägezahn-Chirps direkt nacheinander, was zu einem Radar-Frame mit Zeitdauer T_f zusammengefasst wird. Dies ermöglicht die Prozessierung durch eine zweidimensionale Fast-Fourier-Transformation (FFT) zur separaten Verarbeitung von Distanz und Geschwindigkeit [117]. So kann eine Vielzahl an Objekten mit nur einem Radarscan oder Radarframe detektiert werden. Die Chirps eines Radarframes weisen dabei Zeitdauern von typischerweise 10 – 100 µs auf und benötigen im Vergleich zu traditionellen FMCW-Radarsensoren Sampling-Raten größer 1 MHz.

Abbildung 2.1 veranschaulicht die in dieser Arbeit genutzte Prozessierung [59, 167]. Dabei werden zunächst die AD-Wandler-Samples aller Chirps eines Frames zu einem Block zusammengefasst. Anschließend wird eine 2D-FFT an-

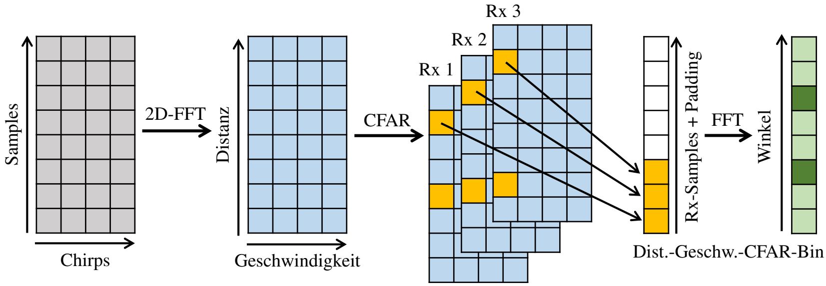


Abbildung 2.1: Veranschaulichung der eingesetzten Fast-Chirp-FMCW-Verarbeitung basierend auf [167]. Die Chirp-Signale werden zunächst digitalisiert, was den grauen Block liefert und anschließend erfolgt die skizzierte Prozessierung. Das Beispiel veranschaulicht ein Radar-Frame bestehend aus vier Chirps und drei Empfangsantennen.

gewendet, was zum Distanz-Geschwindigkeitsblock führt. Für die maximale messbare Distanz d_{max} und die Distanzauflösung (Resolution) d_{res}, also die minimale Distanz, um zwei Objekte voneinander trennen zu können, folgt nach [59, 117]:

$$d_{\text{max}} = \frac{T_c\ c\ f_s}{4B} = \frac{T_c\ c\ f_{\text{adc}}}{2B}, \qquad d_{\text{res}} = \frac{c}{2B}. \tag{2.15}$$

Dabei beschreibt B die Bandbreite des Chirps und f_{adc} die Samplingrate des AD-Wandlers. Die maximale messbare Geschwindigkeit v_{max} sowie die Geschwindigkeitsauflösung, also die minimale Geschwindigkeitsdifferenz, um zwei Objekte mit gleicher Distanz voneinander trennen zu können, ist nach [59, 117] durch

$$v_{\text{max}} = \frac{\lambda}{4T_c}, \qquad v_{\text{res}} = \frac{\lambda}{2T_f} \tag{2.16}$$

gegeben. Dabei beschreibt λ die mittlere Wellenlänge und T_c die Zeitdauer eines Chirps sowie T_f die Zeitdauer eines Radar-Frames.

Die Detektion von Objekten im Ergebnis der zweidimensionalen FFT wird mittels eines Constant-False-Alarm-Rate (CFAR)-Detektors durchgeführt [59]. Dieser schätzt die Charakteristik der vorliegenden Störungen und adaptiert damit den Detektionsschwellwert, um eine konstante Falsch-Alarm-Rate zu

gewährleisten [91]. Ergebnis des CFAR-Detektors sind Punkte innerhalb des Distanz-Geschwindigkeitsblocks, wie durch die gelben Quadrate in Abbildung 2.1 skizziert.

Moderne Sensoren, wie die in dieser Arbeit genutzten, sind mit mehreren Sende- und Empfangsantennen ausgestattet, um zusätzlich zur Distanz und Geschwindigkeit auch die Richtung des Objektes zu schätzen. Dies wird meistens durch Time-Division-Multiple-Access (TDMA) der Sendeantennen umgesetzt, wodurch sich ein virtuelles Antennen-Array ergibt [59]. Dann kann über die CFAR-detektieren Distanz-Geschwindigkeit-Bins aller virtuellen Empfangsantennen eine weitere FFT berechnet werden. Die Peaks innerhalb davon entsprechen Objekten mit gleicher Geschwindigkeit und Distanz, in Abbildung 2.1 für zwei Objekte veranschaulicht. Die Berechnung des Winkels erfolgt dabei nach [59] mit Hilfe des Phasenunterschieds $\Delta\phi$ an den Empfangsantennen was zu

$$\theta = \sin^{-1}\left(\frac{\lambda\, \Delta\phi}{2\pi l}\right) \overset{l=\lambda/2}{=} \sin^{-1}\left(\frac{\Delta\phi}{\pi}\right) \tag{2.17}$$

führt und den nichtlinearen Zusammenhang zeigt. So wird die höchste Güte der Winkelberechnung bei $\Delta\phi = 0°$ erreicht. Unter der Annahme dass die Antennen, wie üblich, den Abstand von $\lambda/2$ aufweisen, ergibt sich die Winkelauflösung, also die Fähigkeit zwei Objekte mit gleicher Distanz und Geschwindigkeit voneinander zu trennen zu

$$\theta_{\text{res}} = \frac{2}{N_{\text{Rx-Antennen}}} \tag{2.18}$$

und hängt damit von der Anzahl an Empfangsantennen $N_{\text{Rx-Antennen}}$ ab. Die Winkelberechnung wird für Azimut θ_A und Elevation ϕ_E durchgeführt und mit Hilfe der Distanz r kann die Position in Kartesischen Koordinaten des Radarsensors mit

$$x^r = r\sin(\theta_A)\cos(\phi_E), \quad y^r = r\sin(\theta_A)\sin(\phi_E), \quad z^r = r\cos(\theta_A) \tag{2.19}$$

aufgestellt werden. Neben den Kartesischen Koordinaten wird auch die Doppler-Geschwindigkeit sowie das Signal-zu-Rausch-Verhältnis zu einem Radarscan-Punkt zusammengefasst. Die Summe aller Radarscan-Punkte eines Radar-Frames ergibt den Radarscan.

2.4 Grundlagen des maschinellen Lernens

Das Interesse am maschinellen Lernen ist mit der enorm gestiegenen Verfügbarkeit von leistungsfähigen Computern in der letzten Dekade sehr stark gestiegen. Eines der verbreitetsten Techniken ist das Deep-Learning, das beispielsweise zur Spracherkennung, Interpretation von Bildern und vielfältigen Klassifizierungsaufgaben eingesetzt wird [108]. Dabei basiert Deep-Learning auf neuronalen Netzen, die aus einer Vielzahl an Schichten und Parametern bestehen [108]. Mit Deep-Learning können sehr komplexe Probleme ohne Modellwissen auf Basis von großen Datensätzen gelöst werden. Für einen Durchbruch bei der Bildverarbeitung haben Convolutional-Neural-Nets (CNNs), eine Art von Deep-Learning gesorgt. Das Ziel von CNNs ist es, Merkmale höherer Ordnung mittels Faltungsoperationen zu lernen [108].

2.4.1 Personendetektion und Posenschätzung

Personendetektion beschreibt die Detektion von Menschen in Bildern mittels rechteckiger Bounding-Boxen. Dafür eigenen sich CNNs sehr gut und können grob in zwei Gruppen eingeteilt werden. Region-based-CNNs besitzen zwei Netze, eines zur Erzeugung von möglichen Objekt-Bounding-Boxen und eines zur Evaluation [31,115]. Single-Shot-CNNs dagegen nutzen ein einzelnes Netz für beide Aufgaben [114,125,157]. Während Region-based-CNNs in der Regel genauer sind, erreichen Single-Shot-Netze deutlich schnellere Laufzeiten und werden daher in dieser Arbeit genutzt.

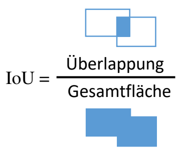
Die Posenschätzung beschreibt die Detektion und Zuordnung von menschlichen Körperteilen in einem Bild, also beispielsweise die Bildkoordinaten des linken Knies oder der rechten Schulter. CNNs erreichen auch bei der Posenschätzung deutlich bessere Ergebnisse als traditionelle Methoden [21]. Eine Einteilung kann beispielsweise in Top-Down- und Bottom-Up-Ansätze vorgenommen werden [21]. Top-Down-Ansätze erstellen zunächst potentielle Bounding-Boxen von Personen und führen dann die eigentliche Posenschätzung durch [21]. Bottom-Up-Ansätze dagegen detektieren Körperteile im gesamten Bild und verbinden dann diese zu Personen [25, 28, 106]. Während Top-Down-Ansätze die höheren Genauigkeiten erreichen, weisen Bottom-Up-Ansätze geringere Laufzeiten auf und werden daher in dieser Arbeit eingesetzt.

2.5 Evaluationsmetriken

Zur Evaluation von Navigationsfiltern werden die folgenden zwei Metriken eingesetzt. Liegt eine Referenz-Trajektorie vor, wird eine Evaluation aller Zeitpunkte der Referenz vorgenommen. Hierzu wird zunächst die geschätzte Trajektorie zur Referenz mit einem *pos-yaw*-Alignment [145] ausgerichtet. Dabei wird mittels eines Least-Squares-Schätzers die translatorische Verschiebung sowie der Gierwinkel bestimmt, der beide Trajektorien unter Minimierung des quadratischen Fehlers zueinander ausrichtet [145]. Auf Basis der ausgerichteten Trajektorien wird dann der translatorische 3D-Fehler der Position von jedem Zeitschritt aufgestellt und der Root-Mean-Square-Fehler berechnet. Dies resultiert in dem weitverbreiteten **Absolute-Translation-Error (ATE)** [159]. Die Berechnung des ATE wurde mit der Toolbox *rpq_trajectory_evaluation* [159] durchgeführt. Liegt für den Datensatz keine Referenz vor und ist die Startposition identisch mit der Endposition, kann der finale Positionsfehler berechnet werden. Die Positionsfehlernorm geteilt durch die Trajektorienlänge ergibt den **relativen finalen Positionsfehler**, der zur Evaluation von Odometrie-Ansätzen weitverbreitet ist.

Ziel dieser Arbeit ist die Entwicklung von Ansätzen zur Navigation und Kartierung in Echtzeit. Zur Evaluation der Ansätze wird daher der **Echtzeitfaktor** genutzt, was die Datensatzlänge geteilt durch die benötigte Verarbeitungszeit ist. Ein Echtzeitfaktor größer als eins bedeutet, dass der Ansatz schneller als Echtzeit verarbeitet werden kann und daher die Anforderung erfüllt.

Zur Evaluation von Detektion und Klassifikation auf Basis von maschinellem Lernen, werden die **Inersection-over-Union (IoU)** und die zwei üblichen, auf der Konfusionsmatrix beruhenden, Metriken **Precision** (Relevanz) und **Recall** (Sensitivität) verwendet, wie in Abbildung 2.2 illustriert ist.



	Vorhergesagt Positiv	**Vorhergesagt Negativ**
Tatsächlich Positiv	True Positive (TP)	False Negative (FN)
Tatsächlich Negativ	False Positive (FP)	True Negative (TN)

$$\text{Precision} = \frac{TP}{TP+FP}$$

$$\text{Recall} = \frac{TP}{TP+FN}$$

Abbildung 2.2: Skizzierung der Intersection over Union (IoU) basierend auf [118] (links) sowie der Konfusionsmatrix (Mitte) mit Definition von Precision und Recall (rechts).

3 Hardwareaufbau

In diesem Kapitel wird der entwickelte Hardwareaufbau beschrieben. Dieser besteht aus einer neu entwickelten Multisensorplattform sowie einer neu entwickelten Flugplattform, auf welche die Multisensorplattform integriert werden kann. Damit kann der gleiche Sensoraufbau sowohl für getragene Datensätze als auch für Testflüge genutzt werden. Die Multisensorplattform wird im ersten und die Flugplattform im zweiten Abschnitt diese Kapitels vorgestellt.

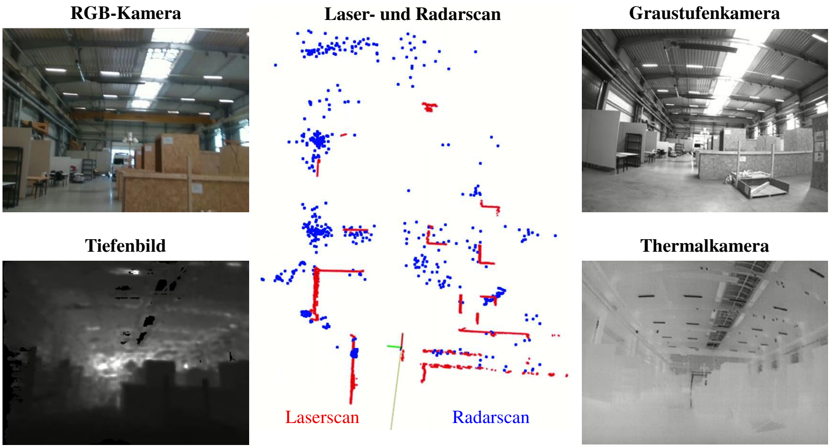
3.1 Multisensorplattform

Im Folgenden wird zunächst die Auswahl der Sensorik begründet. Anschließend erfolgt die detaillierte Beschreibung der entworfenen Multisensorplattform samt aller Sensoren sowie der Rechenplattformen. Abschließend wird die Kalibrierung der Sensoren dargelegt.

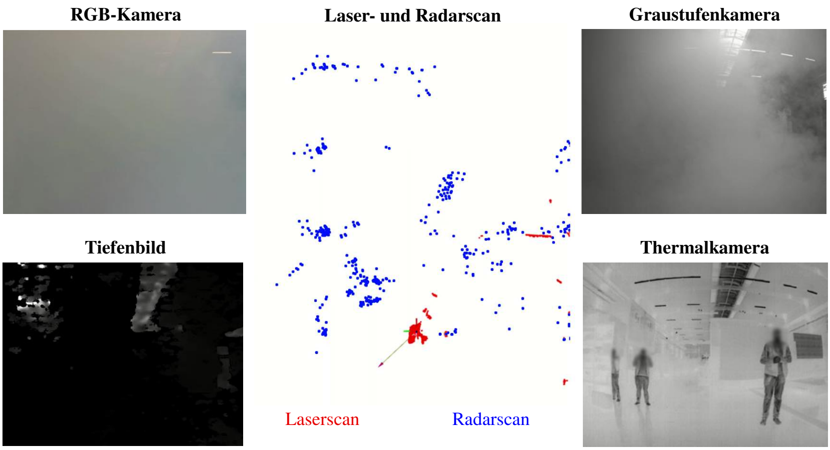
3.1.1 Auswahl von Sensormessprinzipien

Die Auswahl der Sensormessprinzipien wurde mit dem Fokus auf die Ziele der Arbeit, der Navigation und Kartierung auch in schlechten Sichtbedingungen, durchgeführt. Hierfür ist ein Vergleich von auf dem Fluggerät einsetzbarer Sensorik in Abbildung 3.1a für gute und in Abbildung 3.1b für schlechte Sichtbedingungen exemplarisch dargestellt.

In guten Sichtbedingungen wird die Umgebung durch die RGB- und Graustufenkamera sowie dem Tiefenbild und Laserscan sehr gut abgebildet. Die Thermalkamera liefert im Vergleich zur Graustufen-Kamera weniger detaillierte Umgebungsinformationen auf Grund von geringen Temperaturgradienten in dieser Szene. Auch der Radarscan, hier zusammengesetzt aus drei Radarsensoren wie in Abbildung 3.3 verbaut, ist weniger detailliert als der Laserscan.



(a) Gute Sichtbedingungen



(b) Schlechte Sichtbedingungen

Abbildung 3.1: Vergleich von auf dem Fluggerät einsetzbarer Sensorik bei guten Sichtbedingungen (oben) und schlechten Sichtbedingungen (unten). Der Radarscan und die Thermalbilder werden im Gegensatz zum Laserscanner und der RGBD-Kamera (RGB-und Tiefenbild) von den schlechten Sichtbedingungen nicht beeinflusst.

Hier ist die Draufsicht des 3D-Radarscan gezeigt, daher sind mehr Objekte, wie beispielsweise die Deckenträger, als im 2D-Laserscan zu sehen. Flächige Strukturen, wie zum Beispiel Wände, sind im Laserscan deutlicher zu erkennen als im Radarscan. Beide Scans sind über einen Zeitraum von 1 s akkumuliert. In schlechten Sichtbedingungen, hier durch eine Nebelmaschine erzeugt, zeigen sich deutlich die Schwächen von visuellen Sensoren. Die Kamerabilder der RGB- und Graustufenkamera liefern nur sehr wenig Informationen der Umgebung. Auch das Tiefenbild bildet die Umgebung nicht mehr ausreichend ab. Der Laserscan detektiert auf Grund des Nebels fast ausschließlich die minimale Distanz in allen Richtungen. Im Gegensatz dazu liefert der Radarscan und die Thermalkamera die gleichen Daten wie bei guten Sichtbedingungen. Insbesondere die Personen sind trotz des Nebels im Thermalbild sehr klar zu erkennen. Ziel dieser Arbeit ist es ein Navigationssystem und sowie eine Kartierung zu entwickeln, die auch bei diesen Bedingungen einsetzt werden kann. Daher werden im Rahmen dieser Arbeit Navigation mit Radarsensoren sowie einer Thermalkamera untersucht. Visuelle Sensoren werden jedoch ergänzend eingesetzt, um so einen Vergleich zu etablierten Ansätzen der Navigation und Kartierung bei guten Sichtbedingungen durchführen zu können.

3.1.2 Komponenten und Aufbau

Die Multisensorplattform integriert insgesamt sieben Sensoren. Wichtige Eigenschaften aller Sensoren sind in Tabelle 3.1 aufgeführt. Zentraler Sensor und Taktgeber der Zeitreferenz ist eine inertiale Messeinheit (IMU) der Firma *Analog Devices* vom Typ *ADIS16448*. Diese MEMS IMU des mittleren Preissegments erlaubt eine rein inertial-gestützte Navigation über einen kurzen Zeitraum von wenigen Sekunden. Neben der Messung der Beschleunigungen und Drehraten in allen drei Achsen misst die *ADIS16448* auch den Luftdruck mithilfe von einem integrierten Barometer. Eine absolute Positionsbestimmung in Außenbereichen wird mit einem GNSS-Empfänger vom Typ *LEA-M8T* der Firma *u-blox* bereitgestellt. Neben der fertig prozessierten globalen Position liefert der GNSS-Empfänger auch die Rohdaten aller Satelliten in Form von Pseudorange- und Dopplermessungen. Als visuelle RGBD-Kamera ist die *RealSense D435* Kamera von *Intel* verbaut. Diese integriert eine Infrarot-Stereokamera und Farbkamera sowie einen Infrarot-Laserprojektor in einem kompakten Sensor.

Bezeichnung	Sensortyp	Ausgewählte Kenngrößen
Analog Devices *ADIS16448* [161] Kosten: ca. 700 Euro	Beschleunigungs- sensor	Messbereich: ± 18 g Empfindlichkeit: 0,833 mg/LSB In-Run-Bias-Stabilität: 0,25 mg Messrauschen: 0,23 mg/h/$\sqrt{\text{Hz}}$ Updaterate: 205 Hz
	Drehratensensor	Messbereich: $\pm 1000\,°$/sec Empfindlichkeit: 0,04 °/sec/LSB In-Run-Bias-Stabilität: 14,5 °/h Messrauschen: 0,27 °/h/$\sqrt{\text{Hz}}$ Updaterate: 205 Hz
	Barometer	Messbereich: 300 − 1100mbar Empfindlichkeit: 0,02 mbar/LSB Messrauschen: 0,08 mbar $\approx$ 0,95 m Updaterate: 40 Hz
ublox LEA-M8T [162] Kosten: ca. 80 Euro	GNSS-Empfänger	Akquisitionsempfindlichkeit: −148 dBm Updaterate: 10 Hz
Intel *Realsense D435* [169] Kosten: ca. 180 Euro	Infrarot-Stereo- kamera	Auflösung: 1280 × 720 pixel Sichtbereich: 91° × 65° Bildrate: 30 fps
	RGB-Kamera	Auflösung: 1280 × 720 pixel Sichtbereich: 70° × 42° Bildrate: 30 fps
Teledyne-FLIR *Boson 640* [170] Kosten: ca. 4800 Euro	Wärmebildkamera	Auflösung: 640 × 512 pixel Sichtbereich: 95° × 73° Thermale Empfindlichkeit: 50 mK Bildrate: 20 fps
Texas Instruments *IWR6843ISK-EVM* [166] Kosten: ca. 150 Euro	FMCW-Radar	Frequenzbereich: 60 − 64 GHz Antennen: 3 Tx und 4 Rx mit 7 dBi Gewinn Sichtbereich: 120° Azimut, 30° Elevation Winkelauflösung: 14° Azimut, 58° Elevation Updaterate: 10 Hz
Texas Instruments *IWR6843AOP-EVM* [166] Kosten: ca. 150 Euro	FMCW-Radar	Frequenzbereich: 60 − 64 GHz Antennen: 3 Tx und 4 Rx mit 5 dBi Gewinn Sichtbereich: 120° Azimut, 120° Elevation Winkelauflösung: 29° Azimut, 29° Elevation Updaterate: 10 Hz

Tabelle 3.1: Ausgewählte technische Eigenschaften der integrierten Sensorik

Neben den reinen Bilddaten wird auch ein Tiefenbild der Umgebung mit Hilfe eines Bildverarbeitungsprozessors berechnet. Hierbei wird für jeden Pixel mit Hilfe von Triangulation die Tiefe berechnet. Der Laserprojektor hilft hierbei mittels eines zufälligen Musters auch strukturarme Bereiche mit Tiefeninformationen zu versehen. Die Umfeldwahrnehmung wird mit Hilfe von einer Thermalkamera sowie drei Radarsensoren auch in schlechten Sichtbedingungen umgesetzt. Die Thermalkamera vom Typ *Boson 640* der Firma *Teledyne-FLIR* stellt Wärmebilder in einer hohen Auflösung und Update Rate bereit und weist eine hohe Temperaturempfindlichkeit auf. Es werden Radarsensoren der Firma *Texas Instruments* vom Typ *IWR6843AOP* und *IWR6843RISK* verwendet. Um eine Störung der Radarsensoren untereinander zu verhindern, werden diese mit Time Division Multiple Access (TDMA) betrieben, wobei dies mittels Trigger-Signalen umgesetzt wird. Der *IWR6843AOP* weist einen Öffnungswinkel von 120° in Azimut und Elevation auf und integriert die drei Sende- und vier Empfangsantennen auf dem Radarchip. Der *IWR6843RISK*, hat 120° Öffnungswinkel in Azimut aber nur 30° in Elevation und die Antennen sind separat auf der Platine verbaut. Dadurch werden größere Detektionsreichweiten von Objekten mit gleichem Radarquerschnitt als beim AOP erreicht. Die Konfiguration der Radarsensoren ist in Tabelle 3.2 aufgeführt und wurde basierend auch den Systemgleichungen 2.3.2 ausgelegt. Mit den Radarsensoren sind Distanzmessungen bis zu 54 m möglich, wobei diese durch die FFT-Verarbeitung in 0,1 m Schritte quantisiert ist. Die maximale Geschwindigkeit beträgt 2 m/s und ist in 0,13 m/s Schritte quantisiert. Auf Grund von Limitierungen der on-Chip-Radarverarbeitung muss hierbei ein Tradeoff zwischen der Anzahl an Distanz- versus Geschwindigkeitssamples getroffen werden.

Parameter	Wert
Maximale Distanz	54 m
Quantisierung Distanz	0,10 m
Maximale Geschwindigkeit	2 m/s
Quantisierung Geschwindigkeit	0,13 m/s
Maximale Scangröße	256 Objekte
Frame Zeitdauer T_{Frame}	18,5 ms
Chirp Zeitdauer T_{Chirp}	206 µs
Bandbreite	1,53 GHz

Tabelle 3.2: Konfiguration der Radarsensoren

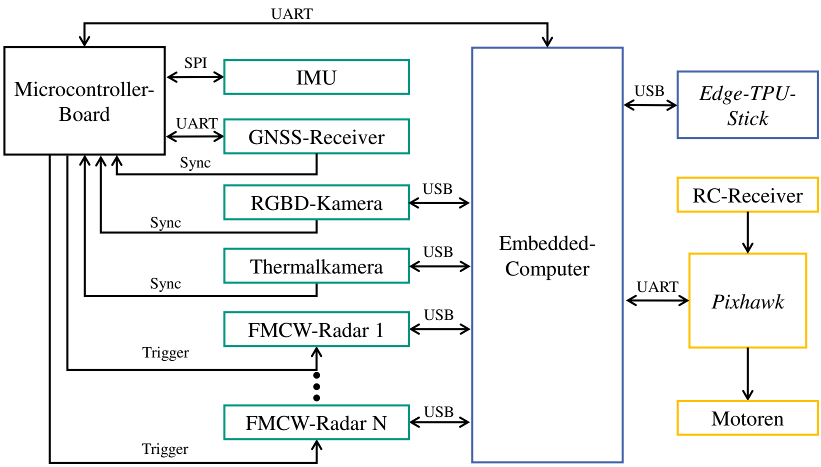


Abbildung 3.2: Blockdiagramm der Multisensorplattform mit Sensorik (grün), Rechenplattformen (blau), Flugplattform-Komponenten (gelb) und Datenfluss (Pfeile).

Abbildung 3.2 zeigt das Blockdiagramm der Multisensorplattform. Die zeitliche Synchronisierung der Sensoren wird dabei mit Hilfe des Microcontroller-Boards implementiert. Dieses empfängt Sync-Signale der *Realsense* und Thermalkamera und sendet Trigger-Signale an die Radarsensoren. Somit ist eine exakte zeitliche Zuordnung der Bilder und Radarscans zum Takt der IMU-Messungen möglich. Das Microcontroller-Board überträgt neben den IMU- und GNSS-Messungen auch die Zeitstempel-Information der anderen Sensoren an den Embedded-Computer. Dieser führt die gesamte online Prozessierung und Zustandsschätzung durch. Rechenintensive neuronale Netze werden auf Spezialhardware, dem *Edge-TPU-Stick*, ausgelagert. Die Flugplattform umfasst neben der Multisensorplattform auch die in gelb dargestellten Komponenten. Die entworfene Multisensorplattform ist in Abbildung 3.3 dargestellt. Bei der mechanischen Integration der Sensoren wurde auf möglichst kurze Hebelarme zwischen den Sensoren sowie auf eine starre Verbindung geachtet. Dadurch ist die Multisensorplattform auch bei Vibrationen und dynamischen Bewegungen mechanisch stabil. Durch die kompakte Bauweise kann die Multisensorplattform auf der Flugplattform, siehe Kapitel 3.2, integriert werden ohne den Sensoraufbau zu verändern.

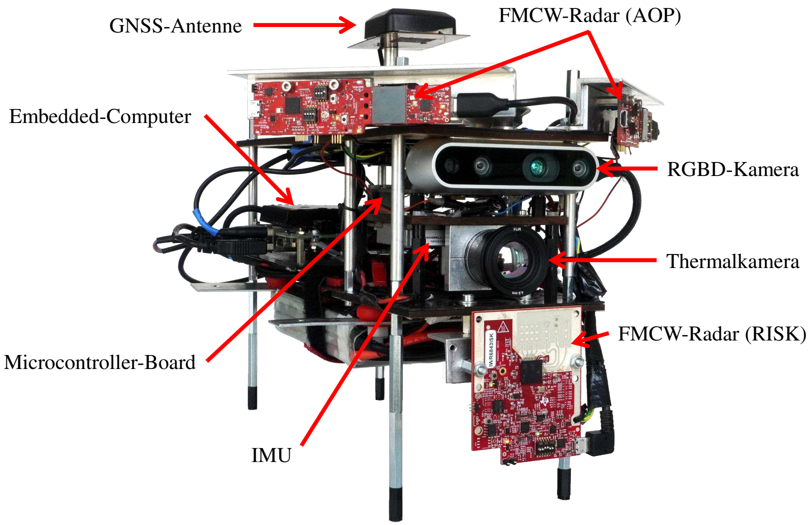


Abbildung 3.3: Multisensorplattform

Im Rahmen dieser Arbeit werden verschiedene Rechenplattformen eingesetzt und evaluiert. So werden zwei verschiedene Embedded-Computer (*NUC-i7* der Firma *Intel* und *Core* der Firma *Up*) verglichen und rechenaufwändige neuronale Netze auf den *Edge-TPU-Stick* von *Google Coral* ausgelagert. Eine Übersicht der Rechenplattformen ist in Abbildung 3.4 dargestellt. Ein detaillierter Vergleich der beiden Embedded-Computer *Intel-NUC-i7* und *Up-Core* ist in Tabelle 3.3 aufgeführt. Das *Intel-NUC-i7* ist zwar etwa doppelt so groß, doppelt so schwer und benötigt doppelt so viel Leistung wie das *Up-Core*, dafür zeigt der Benchmark eine deutlich höhere Performance. Das *Intel-NUC-i7* erreicht hier im Modus mit nur einem Thread (single) mehr als das Fünffache und im Modus mit allen verfügbaren Threads (multi) sogar fast das Siebenfache der Punkte im Vergleich zum *Up-Core*. Beide Rechenplattformen nutzen das Betriebssystem *Ubuntu 16.04* und können auf der Multisensorplattform und Flugplattform eingesetzt werden.

Ziel dieser Arbeit ist es, die gesamte Prozessierung und Kartierung online auf dem Embedded-Computer durchzuführen. Dies ist insbesondere für die Verwendung neuronaler Netze herausfordernd, da hier in der Regel keine Gra-

Parameter	*Intel-NUC-i7* [168]	*Up-Core* [171]
Größe	$10,5 \times 10 \times 3$ cm	$7 \times 6 \times 2{,}4$ cm
Gewicht	220 g	90 g
Leistungsaufnahme	typ. 32 W	typ. 15 W
CPU	Intel Core i7-8650U	Intel Atom x5-Z8350
Maximale CPU Frequenz	4,2 GHz	1,92 GHz
Anzahl Kerne/Threads	4 / 8	4 / 4
Arbeitsspeicher	16 GB	4 GB
Benchmark Punkte [1] Single / Multithread	2246 / 8776	417 / 1269

Tabelle 3.3: Vergleich der beiden Embedded-Computer

fikkarte verbaut ist, die eine Verarbeitung in Echtzeit zulassen würde. Lösung ist die Verwendung eines Hardware-Beschleunigers der per USB angeschlossen werden kann. Hier wird der *Edge-TPU-Stick* von *Google-Coral* eingesetzt. Dieser beinhaltet eine sogenannte Edge-Tensor-Processing-Unit (Edge-TPU), eine anwendungsspezifische integrierte Schaltung zur Ausführung von Inferenzen neuronaler Netze auf Endgeräten mit begrenzten Ressourcen [164]. So können Inferenzaufgaben wie zum Beispiel die Detektion von Personen in Echtzeit ausgeführt werden, ohne hierfür den Embedded-Computer zu belas-

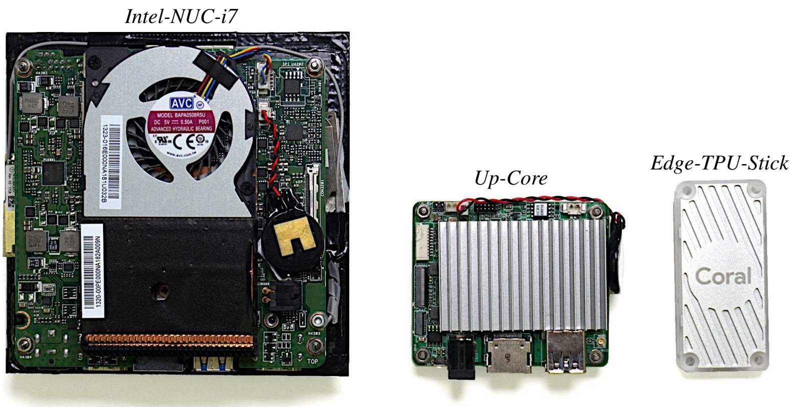


Abbildung 3.4: Eingesetzte Rechenplattformen: *UP-Core* (links), *Intel-NUC-i7* (Mitte) und *Edge-TPU-Stick* (rechts). Die Darstellung der drei Plattformen weist den gleichen Maßstab auf.

Parameter	*Edge-TPU-Stick*
Größe	$6,5 \times 3 \times 0,8$ cm
Gewicht	20 g
Leistungsaufnahme	2 W
Inferenz-Zeit *MobileDet* [157]	20 ms/Bild

Tabelle 3.4: Kenngrößen *Edge-TPU-Stick*

ten. In dieser Arbeit wird das neuronale Netz *MobileDet* zur Personendetektion eingesetzt, was mit Hilfe des *Edge-TPU-Sticks* 20 ms pro Bild benötigt. Wichtige Kenngrößen des *Edge-TPU-Sticks* sind in Tabelle 3.4 aufgeführt.

3.1.3 Kalibrierung

Eine genaue Kalibrierung der Multisensorplattform ist Voraussetzung für eine exakte Navigation. Eine zeitliche Kalibrierung ist bereits durch das Microcontroller-Board umgesetzt. Die extrinsische Kalibrierung, also die Translation und Rotation vom Koordinatensystem des Sensors zum köperfesten Koordinatensystem wurde für die Radarsensoren nur manuell vermessen. Eine exakte Kalibrierung erfolgt online mit dem in Unterabschnitt 5.4.2 erläutertem Ansatz.

Die Kalibrierung der visuellen Kamera (*Intel-Realsense*) wurde mit der Toolbox *kalibr* [44] durchgeführt. Hierbei wird ein Kalibrierboard bestehend aus April-Tag-Markern [150] eingesetzt. Ein Maximum-Likelihood-Schätzer bestimmt die intrinsischen Kameraparameter sowie Translation und Rotation zwischen den beiden Infrarot-Kameras und zwischen der linken Infrarot-Kamera und der IMU. Dies erfolgt auf Basis eines Datensatzes, indem das Kalibrierboard statisch ist und die Multisensorplattform in allen Richtungen translatorisch und rotatorisch bewegt wird.

Das zuvor verwendete visuelle Kalibrierboard kann nicht zur Kalibrierung der Thermalkamera eingesetzt werden, da diese die Umgebungstemperaturen abbildet. Daher wurde ein neues Kalibrierboard entworfen, das sowohl einfach in der Herstellung ist als auch die zuvor verwendete Toolbox *kalibr* [44] nutzen kann. Dies wurde mit einem asymmetrischen Lochmuster gelöst, wie in Abbildung 3.5 dargestellt ist. Typischerweise werden bei der Kalibrierung

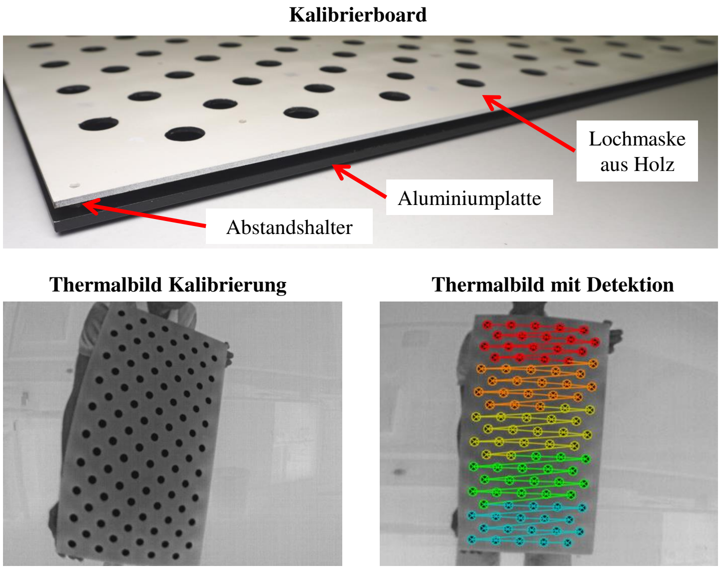


Abbildung 3.5: Kalibrierung der Thermalkamera mit Bild des Kalibrierboards (oben) sowie einem Beispiel mit Detektion (unten).

von Thermalkameras isolierende Masken eingesetzt, die vor einem homogenen Hintergrund montiert werden, sodass sie das Kalibriermuster erzeugen [147]. Dieser Ansatz ist auch hier verfolgt worden. Die Lochmaske wurde dabei mit Hilfe eines Lasercutters in eine Holzplatte geschnitten und mit Kunststoffabstandshaltern vor eine Aluminiumplatte geschraubt, wie in Abbildung 3.5 oben dargestellt ist. Die Aluminiumplatte wurde erwärmt und durch die isolierende Wirkung der Holzplatte ist die Lochmaske auch im Thermalbild sehr gut zu erkennen, siehe Abbildung 3.5 unten links. Nach Mallon et al. [90] wird bei punktförmigen Kalibriermustern ein Kreisdurchmesser von unter 10 Pixel empfohlen, um optimale Kalibriergebnisse zu erhalten. Daher wurde die Lochgröße zu 20 mm gewählt, da so bei einem Abstand von 1-1,5 m der Thermalkamera zum Kalibrierobjekt die Lochgröße im Thermalbild im Bereich zwischen 2 – 8 Pixel liegt.

3.2 Flugplattform

3.2.1 Entwurf und Aufbau

Wie in Abschnitt 1.3 dargelegt, sollen die entworfenen neuen Ansätze zur Navigation und Kartierung unter schlechten Sichtbedingungen mit Testflügen demonstriert werden. Dafür wurde eine neue Flugplattform entworfen, die in Abbildung 3.6 gezeigt ist. Diese soll auch für Flüge in Innenräumen eingesetzt werden können und gleichzeitig in Kombination mit der zuvor beschriebenen Multisensorplattform möglichst lange Flugzeiten erreichen. Daher ergibt sich ein Tradeoff zwischen einer möglichst großen Flugplattform für lange Flugzeit und einer möglichst kleinen Flugplattform für Innenräume. Um auch Türen von 1 m Durchgangsbreite durchfliegen zu können, wurde die maximale Kantenlänge auf 50 cm ausgelegt. Aus Geometriegründen wurde eine x- anstatt einer +-Konfiguration der Motoren gewählt. Die Flugplattform ist in Abbildung 3.6 mit montierter Multisensorplattform dargestellt. Die Motoren sind nach unten ausgerichtet. So ergibt sich ein besseres Sichtfeld der Sensoren, da die Roto-

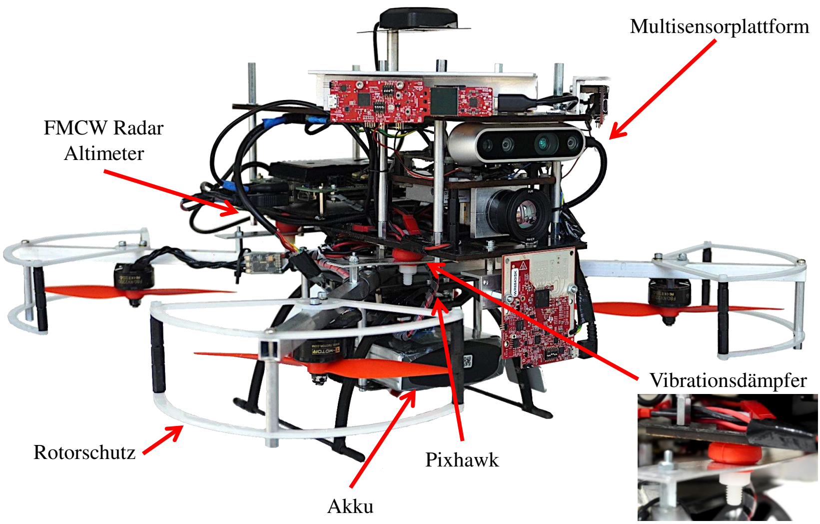


Abbildung 3.6: Flugplattform mit montierter Multisensorplattform

ren und der Rotorschutz nicht in den Kamerabildern zu sehen ist. Zusätzlich beeinflusst der Rahmen die Radarsensoren so nur in einem geringen Maße. Des Weiteren sind nach unten gerichtete Rotoren effizienter als nach oben gerichtete. Das Geschwindigkeitsprofil an klassischen Propellern, wie hier eingesetzt, zeigt unterhalb des Propellers, also in Richtung der Strömung, ein deutlich schmaleres Profil [101]. Daher wirken sich Objekte in der Strömung, wie die Motorausleger, oberhalb des Propellers weniger stark als unterhalb liegende auf die Effizienz aus. Wie in Abbildung 3.6 gezeigt, erfolgt die mechanische Anbindung der Multisensorplattform mittels Vibrationsdämpfern. Diese verringern Vibrationen, die durch Unwuchten der Propeller oder Motoren entstehen. Der etablierte Flugregler *Pixhawk* [165] wird zur Regelung des Quadrokopters eingesetzt. Dieser bietet sehr ausgereifte Regelungsalgorithmen für Quadrokopter mit verschiedenen Flugmodi. So können beispielsweise reine Positions-Sollwerte aber auch komplexere Sollwerte bestehend aus Position, Geschwindigkeit, Beschleunigung, Gierwinkel und Gierrate eingestellt werden. Die Flugplattform mit allen Sensoren, wie in Abbildung 3.6 gezeigt, erreicht mit einem vier-Zellen 6600 mA h Lithium-Polymer-Akku eine Flugzeit von ca. 8 min. Somit können auch längere Demonstrationsszenarien durchgeführt werden. Zur Stützung der Höhe sowie von Start- und Landemanövern, siehe Kapitel 4, ist ein zusätzlicher, nach unten gerichteter FMCW-Radarsensor vom Typ *IWR6843AOP* im hinteren Teil der Multisensorplattform verbaut. Dieser weist im Gegensatz zu den horizontal ausgerichteten Radarsensoren aus Tabelle 3.2 eine geringere maximale Reichweite von 12 m und kleinere Quantisierunsschritte der Distanz (0,04 m) sowie der Geschwindigkeit (0,06 m/s) auf.

3.2.2 Simulationsumgebung

Zur Validierung von Flugführungsalgorithmen wird die realitätsnahe Simulationsumgebung Gazebo [73] eingesetzt. Diese bietet eine umfangreiche Simulation der Physik des Fluggeräts sowie der Sensordaten. Es wird der gleiche Software-Stand des *Pixhawk*-Flugreglers, wie auch auf der realen Flugplattform eingesetzt. So kann effektiv und sicher in Simulation getestet werden, bevor die Flugführungsalgorithmen in Flügen demonstriert werden.

4 Start- und Landemanöver in schwierigen Bedingungen

Vertical Takeoff and Landing Vehicles (VTOLs) sind durch die Fähigkeit senkrecht starten und landen zu können gekennzeichnet. Weit verbreitete Vertreter sind Multikopter, wie in dieser Arbeit eingesetzt, aber auch Schwenkflügler [153, 154]. In diesem Kapitel werden zwei Systeme zur Stützung autonomer Start- und Landemanöver von VTOLs vorgestellt, die autonome Flüge auch in schlechten Sichtbedingungen ermöglichen. Wie im Vergleich der Sensorik in Abschnitt 3.1.1 gezeigt, eignen sich sowohl Radarsensoren als auch Thermalkameras zur Navigation in diesen Bedingungen. Daher werden zwei Systeme im Folgenden entworfen und miteinander verglichen. Es wird in Abschnitt 4.3 das rein Radar-basierte System und in Abschnitt 4.4 ein System basierend auf einer Thermalkamera und einem Radarsensor präsentiert. Das reine Radarsystem besteht aus einer passiven Landeplattform und nutzt nur den Radarsensor, der bereits zur Höhenstützung verbaut ist und wurde in Doer et al. [175] veröffentlicht. Somit wird keine zusätzliche Sensorik für dieses System benötigt. Das System mit Thermalkamera nutzt eine aktive Landeplattform für die Thermalmarker und wurde in Doer et al. [174] veröffentlicht. Dadurch wird eine bessere Performance als beim rein Radar-basierten System mit passiver Landeplattform erreicht. Es wird jedoch mit der nach unten gerichteten Thermalkamera ein zusätzlicher Sensor benötigt. Beide Systeme können in typischen VTOL Szenarien, wie im nächsten Abschnitt motiviert wird, eingesetzt werden. Beide Verfahren werden in Abschnitt 4.5 umfassend miteinander verglichen.

4.1 Szenario

Flüge von VTOLs können typischerweise in die drei Phasen Start, Mission und Landung eingeteilt werden, wie in Abbildung 1.1 in der Einleitung skizziert ist. Beispiele für Missionen können der Transport von Paketen [52, 104], Personen [144] oder Medizin [119, 132] sein. Auch für Such- und Rettungsmissionen [65, 104] sowie Überwachungsmissionen [104, 133] werden VTOLs eingesetzt. Missionen können mittels Wegpunktflügen und GNSS-Stützung ausgeführt werden, was in kommerziellen System realisiert wird. Während der Start- und Landephase wird eine Navigationsstützung benötigt, die es ermöglicht, den Flugpfad horizontal zentriert über dem Landplatz einzuhalten. Da dies mit hoher Genauigkeit erfolgen muss, spricht man auch von Präzisions-Start- und Landemanövern.

Bei Landeplätzen auf Bodenhöhe ergeben sich jedoch oft Bedingungen, die GNSS-basierte Navigation, mit Lokalisierungsfehler unterhalb eines Meters, verhindern. Dies ist auch in Abbildung 1.1 gezeigt. So können Strukturen oder Gebäude, die durch Abschattung oder Mehrwegeausbreitung die GNSS-Genauigkeit erheblich schwächen, zu einem großen Lokalisierungsfehler führen. Des Weiteren stellen aktive Bedrohungen durch Jamming, also dem Stören des GNSS-Signals oder aber Spoofing, der Überlagerung mit einem verfälschten GNSS-Signal, ein erhebliches Sicherheitsproblem dar. Insbesondere autonome unbemannte Flugsysteme benötigen während dieser Phase eine robuste und sichere Navigationsstützung. Zusätzlich soll der Flugbetrieb auch in erschwerten Sichtbedingungen, verursacht zum Beispiel durch Rauch, Nebel, Dunkelheit oder Blendung, sicher ermöglicht werden. Daher können die weit verbreiteten visuellen Navigationsansätze nicht eingesetzt werden, wie in Abschnitt 3.1.1 untersucht ist. Eine passende Navigationsstützung, die diesen Anforderungen gerecht wird, wird im Folgenden durch zwei Systeme realisiert [174, 175]. Sowohl Radar-Sensoren als auch Thermalkameras sind durch die motivierten schlechten Sichtbedingungen kaum beeinflusst, wie in Abbildung 3.1 aus Abschnitt 3.1.1 analysiert.

Konkret stellt sich ein Flug mit den vorgeschlagenen Systemen wie folgt dar: Während des Starts erfolgt die Lokalisierung relativ zur Landeplattform mittels eines der beiden entwickelten Systeme. Ab einer ausreichenden Flughöhe wird die Mission gestartet und die Lokalisierung beispielsweise mittels GNSS durchgeführt. Zur Landung wird zunächst ein Wegpunkt oberhalb der Lan-

deplattform in ausreichender Höhe angeflogen. Sobald die Landeplattform detektiert ist, erfolgt die Lokalisierung relativ dazu und die Präzisionslandung wird ausgeführt.

4.2 Verwandte Arbeiten

Systeme zum präzisen Starten und Landen von VTOLs sind gegenwärtig ein aktives Forschungsfeld. Präzisionslandesysteme basierend auf Real Time Kinematic (RTK)-GPS sind bereits etabliert und können auch auf VTOLs eingesetzt werden [37]. Neben RTK bieten sich auch Dienste wie Precise Point Positioning (PPP) an, da diese keine selbst betriebene Bodenstation in der Nähe benötigen und Zentimetergenauigkeit weltweit in Open Sky Bedingungen erreichen [103]. Wie jedoch im Szenario aus Abschnitt 4.1 erläutert, stellt eine GNSS-basierte Stützung bei Start- und Landemanöver auf Grund der unter Umständen limitierten Verfügbarkeit und Sicherheitsrisiken keine zufriedenstellende Lösung dar.

In der konventionellen Luftfahrt kommen daher Systeme wie das Instrument Landing System (ILS) oder das Microwave Landing Sytem (MLS) zur Stützung zum Einsatz. Eine Analyse zur Verwendbarkeit des MLS auf konventionellen Helipads hat gezeigt, dass die benötigte Fläche zu groß ist, um einen realistischen Einsatz in urbanen Umgebungen zu ermöglichen [109]. Aufgrund der Kosten, Größe und Gewicht der notwendigen Komponenten können solche Systeme im motivierten Szenario aus Abbildung 1.1 nicht eingesetzt werden.

Eine sehr häufig einsetzte Alternative zu GNSS-basierter Navigation ist die visuelle Navigation [62, 92, 156]. Bei diesen Ansätzen werden visuelle Marker in die Landeplattform integriert und mit einer nach unten gerichteten visuellen Kamera detektiert. Hierbei kommen eine Vielzahl an visuellen Markern zum Einsatz [15, 75, 116, 151]. Am häufigsten werden quadratische Marker genutzt, da diese zusätzlich auch eine Marker ID integrieren, wie zum Beispiel durch April Tag [151] oder ArUco [116] umgesetzt. Kreisförmige Marker, zum Beispiel mittels Blobs [15] oder WhyCon [75] sind zwar deutlich effizienter hinsichtlich der Rechenzeit, können aber keine ID integrieren. Um diese Marker-basierten Ansätze auch in Dunkelheit nutzen zu können, bietet sich eine aktive Beleuchtung der Marker an [156]. Die Anwendung dieser Ansätze ist jedoch limitiert durch schlechten Sichtbedingungen wie beispielsweise Nebel

oder Rauch, was die Anwendung für das motivierte Szenario aus Abbildung 1.1 verhindert. Einzig die Nutzung von Thermalmarker, wie durch Khattak [70] gezeigt, ist eine Möglichkeit Marker-basierte Ansätze zu nutzen. Thermalmarker wurden jedoch bisher nicht zur Stützung von Start- und Landemanövern von VTOLs eingesetzt.

Eine Möglichkeit ist die Nutzung von Ultra Wideband (UWB) [24, 30, 36, 143] da dieses Verfahren Bereiche des elektromagnetischen Spektrums nutzt, die durch schlechte Sichtbedingungen nur wenig beeinflusst werden. UWB nutzt dabei Laufzeitmessungen zwischen den UWB-Ankern und UWB-Tags. Jedes UWB Tag bestimmt dabei die Laufzeit zu jedem anderen UWB-Anker. Bei bekannter Position der UWB-Anker kann so mittels Triangulation, ähnlich wie bei GNSS, die Position des UWB Tags geschätzt werden [143]. Auf Grund der Geometrie des Problems, wird jedoch eine recht große Fläche benötigt, auf der die UWB-Anker verteilt werden müssen [24, 36]. Dieses Problem kann mit dem Ansatz von Dobrev [36] umgangen werden, da hier die Distanz als auch der Empfangswinkel in Azimut und Elevation vom UWB-Tag gemessen wird. So kann ein System mit nur einem Anker auf dem Boden sowie einem UWB-Tag auf dem VTOL realisiert werden [36]. Solch ein Ansatz benötigt allerdings spezielle Hardware. Ziel dieser Arbeit ist es jedoch ein System zu entwerfen, das mit kommerziell verfügbaren Produkten realisiert werden kann.

Eine weitere Möglichkeit ist die Nutzung von FMCW Radar, wie bereits anfangs dieser Arbeit in Abschnitt 3.1.1 analysiert. FMCW-Radarsensoren wurden in diesem Kontext bisher jedoch nur zur reinen Höhenstützung eingesetzt [57, 58, 129]. Radar-basierte Höhenstützung hat, wie durch Schartel [129] gezeigt, deutlich bessere Ergebnisse als Laser-basierte Stützung erzielt. Eine Stützung der horizontalen Position mit Hilfe eines nach unten gerichteten FMCW-Radarsensors wurde bisher noch nicht betrachtet.

Die Analyse der verwandten Arbeiten zeigt den Bedarf an neuen Ansätzen zur robusten und sicheren Stützung der Position und Orientierung eines VTOLs bei Start und Landung, die auch für Szenarien mit schlechten Sichtbedingungen eingesetzt werden kann. Dabei wurden im Rahmen dieser Arbeit zwei verschiedene Systeme entworfen und in Doer et al. [174, 175] veröffentlicht. Zunächst wurde der Ansatz Marker zu nutzen zu einer neuartigen Anwendung mit kommerziellen FMCW-Radarsensoren entwickelt [175]. Hierbei werden passive Radar-Reflektoren in die Landeplattform integriert und detektiert. So kann mittels eines FMCW-Radarsensors neben der Höhe auch die horizon-

tale Position sowie Ausrichtung relativ zur Landeplattform gestützt werden. Das Radar-basierte System ist in Abschnitt 4.3 näher beschrieben und evaluiert. Des Weiteren wurde, basierend auf visuellen Markern, der Ansatz Thermalmarker zu nutzten zu einem neuartigen System zu Start- und Landestützung weiterentwickelt [174]. Hierbei werden aktive Thermalmarker in die Landeplattform integriert und mit FMCW-Radar-basierter Höhenstützung kombiniert. Das Thermalkamera-Radar-basierte System wird in Abschnitt 4.4 präsentiert und evaluiert. Ein abschließender Vergleich beider Systeme erfolgt in Abschnitt 4.5.

4.3 Radar-basiertes Start- und Landesystem

Dieser Abschnitt stellt ein rein Radar-basiertes System zur genauen relativen Lokalisierung für Start- und Landemanöver vor und wurde in Doer et al. [175] veröffentlicht. Da FMCW-Radar nur sehr schwer zu stören ist [105], erfüllt die Sensorik die motivierten Sicherheitsanforderungen autonomer Flugsysteme. Zusätzlich ist das System passiv, es werden also an der Landeplattform keine aktiven Elemente benötigt. Dadurch kann diese kostengünstig umgesetzt werden. Der Systemaufbau ist in Abbildung 4.1 dargestellt. Die Landeplattform besteht aus einer 1×1 m großen Plexiglasplatte. Unterhalb der Plexiglasplatte befinden sich die Radar-Reflektoren der Start- und Landeplattform. Diese werden mit Hilfe des nach unten ausgerichteten FMCW-Radarsensors am VTOL detektiert. Im Folgenden wird auf die einzelnen Elemente der Landeplattform näher eingegangen.

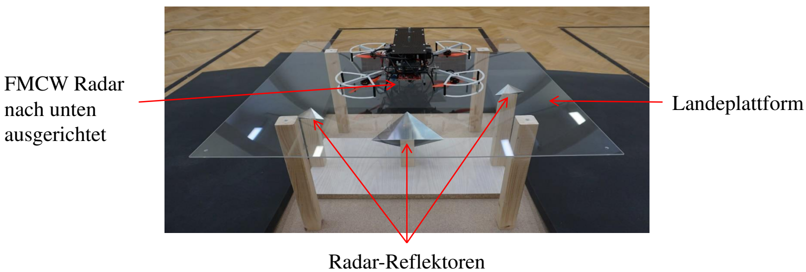


Abbildung 4.1: Radar-basiertes Start- und Landesystem

4.3.1 Radarreflektoren

Radarreflektoren sind Strukturen, die Radarstrahlung erheblich stärker reflektieren als andere Strukturen der Umgebung. Eine Einteilung ist nach [122] in drei Hauptgruppen möglich: Retroreflektoren, dielektrische Linsen oder Retrodirective Arrays. Retroreflektoren weisen dabei die geringsten Kosten auf und sind leicht in der Herstellung. Insbesondere triangulare Winkelreflektoren, im linken Teil von Abbildung 4.2 gezeigt, sind sehr weit verbreitet, da sie einen großen Sichtbereich aufweisen [122]. Nach [122] ist der maximale Radarquerschnitt $\sigma_{\max}$ eines triangularen Winkelreflektors durch

$$\sigma_{\text{RCS,max}} = \frac{4\pi a^4}{3\lambda^2} \tag{4.1}$$

mit der Kantenlänge a sowie der Wellenlänge λ gegeben. Eine Analyse des Sichtbereichs ist wichtig, um die Landeplattform korrekt auslegen zu können. Dies kann mit Hilfe des relativen Radarquerschnitts durchgeführt werden. Dieser zeigt wie hoch die Dämpfung des Signals in Bezug auf den Einfallswinkel ist. Nach [122] folgt für den relativen Radarquerschnitt $\sigma_{\text{RCS,rel}}$ des eingesetzten Winkelreflektors

$$\begin{aligned} \sigma_{\text{RCS,rel}}(\theta_A, \phi_E) \approx & \frac{4\pi}{\lambda^2} a^4 [cos\theta_A + sin\theta_A\,(sin\phi_E + cos\phi_E) \\ & - 2\,(cos\theta_A + sin\theta_A\,(sin\phi_E + cos\phi_E))^{-1}]^2 \end{aligned} \tag{4.2}$$

mit dem Azimutwinkel θ_A und Elevationswinkel ϕ_E aus Abbildung 4.2 links. Hiermit lässt sich die Dämpfung in Bezug auf den Einfallswinkel berechnen, wie in Abbildung 4.2 rechts aufgetragen. Dies zeigt die Abweichung in Azimut und Elevation von der Symmetrieachse, die in Abbildung 4.2 links durch den gestrichelten Pfeil eingezeichnet ist. Die Symmetrieachse liegt bei dem eingesetzten Winkelreflektor bei $\theta_A = 45°$ und $\phi_E = 54{,}75°$ [122]. Es ergibt sich somit ein Sichtbereich von circa 20° bei dem die Dämpfung weniger als 6 dB beträgt.

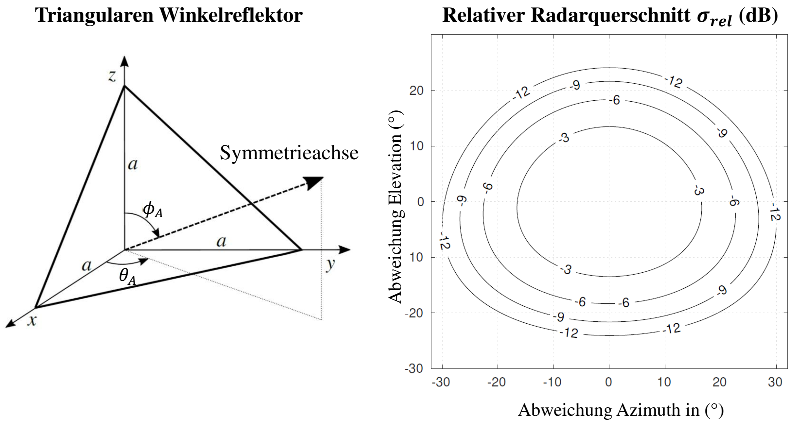


Abbildung 4.2: Triangulare Winkelreflektor (links) mit relativen Radarquerschnitt in Bezug auf die Symmetrieachse (rechts) nach Doer et al. [175].

4.3.2 Entwurf der Start- und Landeplattform

Der Entwurf der Start- und Landeplattform orientiert sich an der Charakteristik des FMCW-Radar sowie der Winkelreflektoren. Nach Vorschrift der EASA muss die Fläche des Landeplatzes frei von Hindernissen sein [39]. Daher müssen die Radarreflektoren unterhalb der Landeplattform integriert werden. Dies erfolgt, wie in Abbildung 4.1 gezeigt, durch eine Plexiglasplatte da diese die elektromagnetische Strahlung des Radarsensors nur geringfügig dämpft. Für eine robuste Detektion, die auch die Schätzung des Gierwinkels ermöglicht, werden mehrere Radarreflektoren eingesetzt.

Konkret besteht die Landeplattform aus drei Winkelreflektoren, siehe Abbildung 4.1. Hierbei werden zwei verschiedene Größen eingesetzt. Der mittlere Reflektor weist eine Kantenlänge von 20 cm und die beiden äußeren eine Kantenlänge von 10 cm auf. Damit ergibt sich unter Verwendung von (4.1) ein Radarquerschnitt von 453 m^2 und 28 m^2. Der mittlere Reflektor ist zentral in der Landeplattform integriert und ermöglicht durch den großen Radarquerschnitt eine robuste Trennung von der Bodenreflexion. Eine Trennung der Reflektoren nur auf Basis des Winkels ist mit dem verwendeten FMCW-Radar (RISK) nicht möglich. Dieser weist eine Winkelauflösung von 14° in Azimut

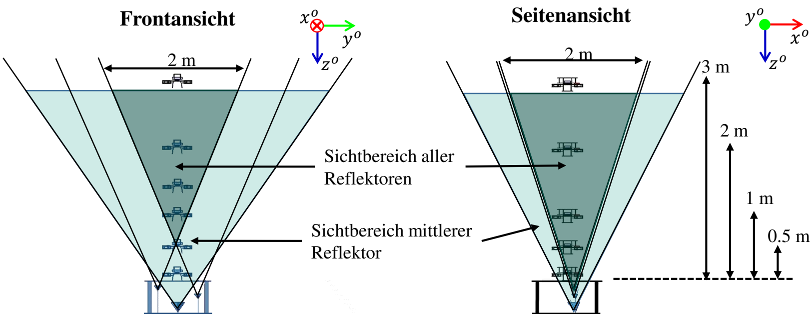


Abbildung 4.3: Analyse der Sichtbereiche in Front- und Seitenansicht nach Doer et al. [175]. Der dunkel grüne Bereich zeigt das Volumen in dem alle Reflektoren im Sichtbereich sind.

und 57° in Elevation auf. Daher wird die Trennung in einzelne Objekte mittels der Distanz realisiert. Da der Radarsensor mit der maximalen Bandbreite von 4 GHz betrieben wird, ergibt sich eine Distanzauflösung von 3,9 cm. Für eine robustere Trennung, weisen die einzelnen Reflektoren einen Höhenunterschied von 12 cm auf. Die Verteilung in der y^o-Achse ist ein Tradeoff zwischen der minimalen Höhe an der alle Reflektoren in Sichtbereich des Radar sind und der Genauigkeit der Lokalisierung. Hier wurde ein Abstand von 40 cm gewählt. Dies maximiert den Abstand im Rahmen der Landeplattformgeometrie und hat alle Reflektoren im Sichtbereich ab einer Höhe von 0,75 m. Eine Darstellung der Sichtbereiche ist in Abbildung 4.3 gezeigt. Gut zu erkennen ist auch die Vergrößerung des Korridors in Abhängigkeit der Höhe. Unterhalb von 0,75 m über der Landplattform befindet sich nur der mittlere Reflektor im Sichtbereich. Da dieser jedoch einen sehr großen Radarquerschnitt aufweist, ist eine robuste Detektion möglich.

4.3.3 Detektion der Landeplattformreflektoren

Dieser Abschnitt beschreibt die Detektion bzw. die Assoziierung der drei Radarreflektoren mit den Objekten des Radarscans. Der einfachste Weg zur De-

tektion wäre mit Hilfe des Radarquerschnitts. Umformen von (2.14) nach dem Radarquerschnitt σ_{RCS} ergibt:

$$\sigma_{\mathrm{RCS}} = \frac{(4\pi)^3 R^4}{G_s G_e \lambda^2} \frac{P_e}{P_s}. \tag{4.3}$$

Während die Gewinne der Sende- und Empfangsantennen G_s und G_e sowie die Sendeleistung P_s und Distanz R bekannt sind, erlaubt der verwendete Sensor keine absolute Messung der Empfangsleistung P_e. Daher wird die Detektion der Landeplattform Reflektoren durch deren eindeutige Geometrie durchgeführt.

Hierfür wird der Radarscan zunähst bezüglich Distanz, Signal zu Rauschverhältnis, Azimut- und Elevationswinkel gefiltert. Dies entfernt unwahrscheinliche Messungen und beschleunigt die nachfolgende Detektion. Als nächstes werden alle Objekte $\boldsymbol{p}_o^r$ des Radarscans in das stabilisierte Koordinatensystem $\{\}^s$ transformiert:

$$\boldsymbol{p}_o^s = \mathbf{C}_b^s \mathbf{C}_r^b \boldsymbol{p}_o^r. \tag{4.4}$$

Dabei wird die Transformation $\mathbf{C}_b^s$ mit der aktuellen Navigationslösung berechnet und die Rotation des Radarsensors bezüglich des körperfesten Koordinatensystems $\mathbf{C}_r^b$ als gegeben betrachtet. Es wird angenommen, dass die Landeplattform zur Erdbeschleunigung ausgerichtet ist. Somit kann die Assoziation der einzelnen Reflektoren mittels der x-, y- und z-Komponente der transformierten Positionen $\boldsymbol{p}_o^s$ erfolgen. Falls eine Detektion nicht möglich ist, beispielsweise weil die geforderte Geometrie nicht mit einer erforderlichen Güte erreicht worden ist, und das Objekt mit dem stärksten Signal weniger als 0,75 m entfernt ist, wird dieses als mittlerer Marker klassifiziert. Auf Basis der Detektion kann eine relative Lokalisierung durchgeführt werden, wie im nächsten Abschnitt erläutert.

4.3.4 Navigationsstützung

Mit der assoziierten Reflektor Messung $\boldsymbol{p}^r_{m,\{L,M,R\}}$ sowie der korrespondierenden Position des Reflektors $\boldsymbol{p}^o_{\{L,M,R\}}$ im lokalen Koordinatensystem der Landeplattform kann die Messgleichung

$$\boldsymbol{p}^o_{ob} = \mathbf{C}^o_b \left(\mathbf{C}^b_r \boldsymbol{p}^r_{m,\{L,M,R\}} - \boldsymbol{l}^b_{br} \right) - \boldsymbol{p}^o_{\{L,M,R\}} \tag{4.5}$$

aufgestellt werden Hierfür wird jedoch die Transoformation Lage $\mathbf{C}^o_b$ benötigt. Während der Roll- und Nickwinkel mit dem Navigationsfilter mit Hilfe der Erdbeschleunigung geschätzt werden kann, gilt dies für den Gierwinkel nur unter Verwendung von globaler Stützinformation beispielsweise durch GNSS. Durch die eindeutige Geometrie der Landeplattformreflektoren kann der Gierwinkel jedoch wie folgt bestimmt werden. Es folgt zunächst die Transformation in das stabilisierte Koordinatensystem mittels (4.4). Da die Landeplattform das lokale Koordinatensystem aufspannt, kann der lokale Gierwinkel ψ_{ob} mit den assoziierten Reflektormessungen in drei Kombinationen aufstellt werden:

$$\psi_{ob,ML} = \text{atan2}(d^s_{x,ML},\ d^s_{y,ML}), \tag{4.6}$$

$$\psi_{ob,RL} = \text{atan2}(d^s_{x,RL},\ d^s_{y,RL}), \tag{4.7}$$

$$\psi_{ob,RM} = \text{atan2}(d^s_{x,RM},\ d^s_{y,RM}), \tag{4.8}$$

mit dem Verbindungsvektor $\boldsymbol{d}^s_{AB}$:

$$\boldsymbol{d}^s_{AB} = \boldsymbol{p}^s_A - \boldsymbol{p}^s_B. \tag{4.9}$$

Die finale Gierwinkelmessung $\psi^o_{ob,m}$ wird über den Mittelwert aller drei Messungen bestimmt:

$$\psi_{ob,m} = \frac{1}{3} \sum_{i \in \{L,M,R\}} \psi_{ob,i}. \tag{4.10}$$

Da somit der Gierwinkel $\psi_{ob,m}$ bekannt ist, kann die Rotationsmatrix $\mathbf{C}_b^o$ aufgestellt und die Messung der globalen Position $\boldsymbol{p}_{ob,m}^o$ per Mittelung aller drei Messungen

$$\boldsymbol{p}_{ob,m}^o = \frac{1}{3} \sum_{i \in \{L,M,R\}} \left(\mathbf{C}_b^o \left(\mathbf{C}_r^b \boldsymbol{p}_{m,i}^r - \boldsymbol{l}_{br}^b \right) - \boldsymbol{p}_i^o \right) \tag{4.11}$$

berechnet werden. Befindet sich das VTOL nur im Sichtbereich des mittleren Reflektors, kann eine Gierwinkelmessung nicht durchgeführt werden. Da dies jedoch nur über sehr kurze Zeiträume beim Start und der Landung auftritt, kann hier eine rein inertiale Stützung des Gierwinkels durchgeführt werden.

Um die Genauigkeit der Lokalisierung weiter zu verbessern, wird eine Fusion mit Inertialsensorik in einem Kalman-Filter durchgeführt. Dieser fusioniert die Positionsmessung $\boldsymbol{p}_{ob,m}^o$ sowie die Gierwinkelmessung $\psi_{b,m}^o$. Outlier werden durch einen Mahalanobisdistanz-Test, wie im Anhang A.1 erläutert, entfernt.

4.3.5 Ergebnisse

Es werden im Folgenden zwei Reihen von Flugexperimenten präsentiert. Zum einen Testflüge mit einem Motion-Capture System zur Evaluation der Genauigkeit und zum anderen realistische Realflugtests in schlechten Sichtbedingungen. Hierfür wurde das Indoorflugsystem leicht modifiziert, wie im Folgenden kurz erläutert.

Hardwareaufbau

Für die folgenden Flugexperimente wurde die Flugplattform aus Abschnitt 3.2 leicht modifiziert, wie in Abbildung 4.4 links dargestellt. Der RISK Radarsensor wurde so montiert, dass er nach unten zeigt. Zusätzlich zeigt auch eine visuelle Kamera nach unten. Diese wird jedoch nur zur Visualisierung verwendet. Als Embedded-Computer wird das *Up-Core* eingesetzt.

Experimentelle Ergebnisse mit Motion-Capture

Zur Evaluation der Lokalisierungsgenauigkeit wurden Testflüge im Bereich von $1-2$ m oberhalb der Landeplattform im Motion-Capture Lab durchgeführt. Ein Überblick des Versuchsaufbaus ist in Abbildung 4.4 auf der rechten Seite gezeigt. Mit Hilfe des Motion-Capture Systems liegt dabei eine exakte Referenz vor.

Abbildung 4.5 zeigt die Flughöhe (oben) sowie die Fehlernorm der Position (Mitte) sowie den absoluten Gierwinkelfehler (unten). Im Bereich zwischen 12 – 25 s wurde bei einer konstanten Höhe die horizontale Position variiert. Eine Evaluation kann durchgeführt werden, indem die Messungen der Position $\boldsymbol{p}^{o}_{ob,m}$ sowie des Gierwinkels $\psi_{ob,m}$ mit der Referenz verglichen werden. Trotz vereinzelter Outlier beim Positionsfehler beispielsweise bei 3,5 s bleibt der Fehler mit einer mittleren Fehlernorm von 6 cm sehr klein. Der Gierwinkel zeigt dagegen größere Fehler mit einem mittleren Fehlerbetrag von 3,8°. Dieser wird durch den, im Verhältnis zur y^o-Achse, um circa den Faktor drei größeren Fehler in der x^o-Achse verursacht (4,7 cm versus 1,6 cm). Während der Azimut Winkel durch ein virtuelles Antennenarray von acht Antennen sehr genau

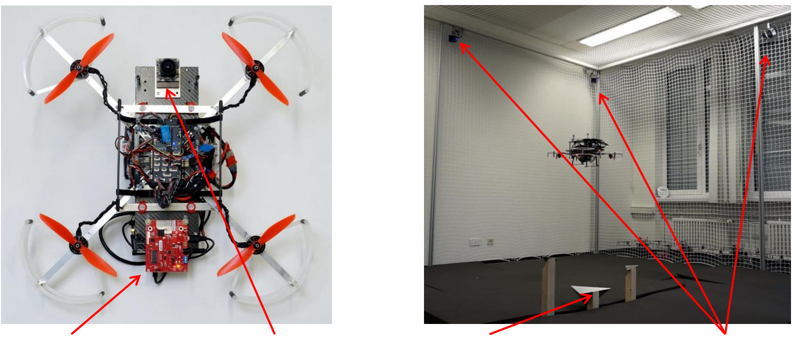


Abbildung 4.4: Modifizierte Flugplattform (links) und Testflug für Evaluation mit Motion-Capture (rechts) nach Doer et al. [175].

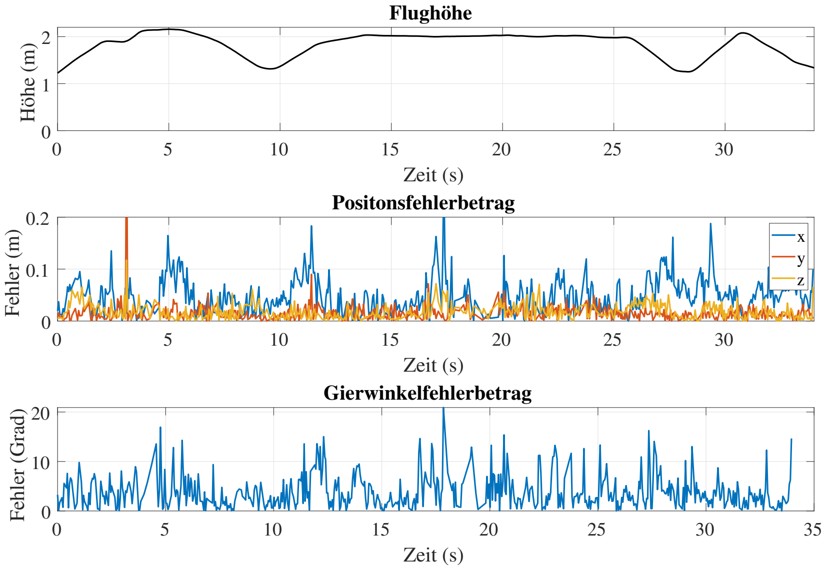


Abbildung 4.5: Evaluation des absoluten Positionsfehlers (oben) und Gierwinkelfehlers (unten) mittels Motion-Capture Referenz.

geschätzt werden kann, ist die Genauigkeit des Elevationswinkel auf Grund von nur zwei Antennen eingeschränkt. Daraus folgt eine schlechtere Schätzung der x^o-Achse, was sich auch in der Gierwinkelmessung zeigt. Die Genauigkeit der z^o-Achse wird im Wesentlichen durch die Schrittweite der Distanz FFT, siehe Unterabschnitt 2.3.2, bedingt. Der FMCW-Radarsensor wurde mit der besten Distanzgenauigkeit parametrisiert, was zu einer Schrittweite von 4 cm führt. Durch die Mittelung der drei Messungen ergibt sich eine Verbesserung des z^o-Achsenfehlers, der hier einen mittleren absoluten Fehler von 1,7 cm annimmt. Diese Ergebnisse zeigen, dass die Navigationsstützung im kritischen Bereich direkt oberhalb der Landeplattform sehr genau ist, was eine äußert präzise Landung ermöglicht, wie im nächsten Abschnitt demonstriert.

Experimentelle Ergebnisse Sporthalle

Eine Demonstration des Radar-basierten Systems zur Start- und Landestützung wird mittels vier Testflügen in schlechten Sichtbedingungen demonstriert. Diese wurden in einer Sporthalle durchgeführt, wobei schlechte Sichtbedingen mit einer Nebelmaschine erzeugt worden sind. Zwei Aufnahmen bei einer Höhe von circa 0,5 m und 3 m sind in Abbildung 4.6 gezeigt. Trotz der schlechten Sichtbedingungen waren alle Testflüge erfolgreich.

Die Testflüge wurden vollständig autonom durchgeführt wobei die folgende Flugführung, motiviert durch das Szenario aus Abbildung 1.1, angewendet wurde. Start bis in eine definierte Höhe, anschließend wird die Mission durch Hovern über einen Zeitraum von 10 s durchgeführt und anschließend die Präzisionslandung ausgeführt. Die Landung erfolgt zunächst mit einer Geschwindigkeit 1 m/s. Bei einer Höhe von 1 m oberhalb der Landeplattform erfolgt eine Hoverphase, bis die horizontale Abweichung unterhalb einer definierten Schwelle ist. Anschließend erfolgt die Landung. Es wurden die Höhen 2 m und 3 m als Zielhöhe evaluiert.

Abbildung 4.7 zeigt in der oberen Reihe die Radarscans mit Kamerabildoverlay und in der unteren Reihe das Ergebnis der Landeplattform-Detektion. Dabei zeigt die linke Spalte einen Flug bei guten Sichtbedingungen. Trotz der teils



Abbildung 4.6: Testflüge in schlechten Sichtbedingungen bei einer Höhe von 0,5 m (links) und 3 m (rechts). Die Flugplattform ist mit der gestrichelten Box markiert.

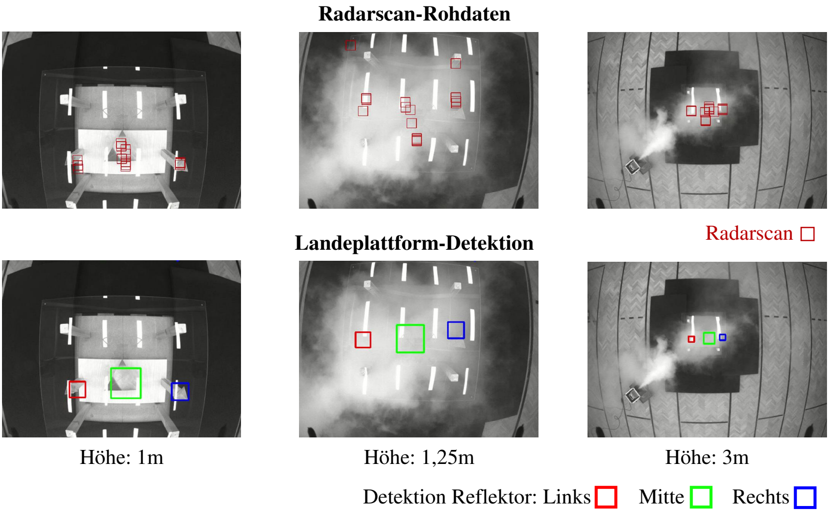


Abbildung 4.7: Kameraoverlay mit den Radarscans (oben) sowie dem Ergebnis der Landeplattformdetektion (unten) nach Doer et al. [175].

verstreuten Messungen des Radarscans, kann die Landeplattform unabhängig der Sichtbedingungen korrekt detektiert werden.

Eine Analyse der geschätzten Position von Flug 4 ist in Abbildung 4.8 dargestellt. Hierbei sind die Positionsmessungen des Radars in blau dargestellt und das Ergebnis der Fusion mit Inertialsensorik in Rot. Die geschätzte Position zeigt einen sehr glatten Verlauf, wobei vereinzelte Outlier entfernt wurden.

4.4 Thermalkamera-basiertes Start-und Landesystem

In diesem Abschnitt wird ein System für Start und Landemanöver mit Hilfe von einer Thermalkamera und einem FCMW Radar realisiert. Dieses System wurde in Doer et al. [174] veröffentlicht. Im Gegensatz zum rein Radar-basierten System aus Abschnitt 4.3, bietet die Kombination beider Sensoren eine höhere

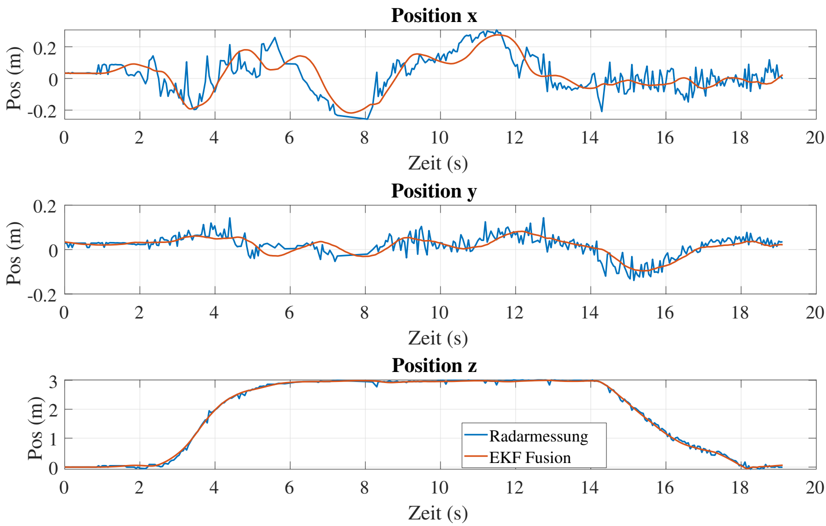


Abbildung 4.8: Plot der Radar-Positionsmessungen (blau) und der Kalman-Filter-Position von Flug 4 nach Doer et al. [175].

Genauigkeit und weitaus größere Detektionsreichweiten. Obwohl Thermalkameras hochpreisig sind, werden sie bei vielen Anwendungen, zum Beispiel bei der Fernaufklärung von Vegetation [11] oder bei Such- und Rettungsmissionen [38], auf VTOLs eingesetzt. Daher bietet sich die Verwendung der Thermalkamera auch für Start- und Landemanöver an, da diese bereits auf dem Fluggerät integriert ist. Das entwickelte System ist in Abbildung 4.9 abgebildet. Es besteht aus einer 1×1 m großen Plattform in die Thermalmarker integriert sind. Das VTOL, in diesem Fall die Flugplattform, verfügt über eine Thermalkamera sowie einen FMCW-Radar. Beide Sensoren sind dabei nach unten ausgerichtet. Das System ist aktiv, das heißt es wird eine Spannungsquelle benötigt, um die Thermalmarker zu betreiben. Hierfür wird jedoch nur eine geringe Leistung benötigt und limitiert somit mobile Anwendungen nicht. Die im Versuch verwendete Hardware besitzt eine Leistungsaufnahme von 10 W. Die Bestandteile des Start- und Landesystems werden im Folgenden näher beschrieben.



Abbildung 4.9: Thermalkamera-basiertes Start- und Landesystem

4.4.1 Thermalmarker

Die Detektion von Markern in visuellen Bildern wurde bereits sehr umfangreich erforscht. Weit verbreitet sind quadratische Marker, zum Beispiel ArUco [116] oder April Tag [151], da diese Marker auch eine ID integrieren, ähnlich wie ein QR-Code. Unter der Annahme, dass jede Marker ID nur einmal in der Umgebung existiert, kann jede Detektion einem realen Marker eindeutig zugeordnet werden. Im Gegensatz zu den QR-Code Markern, wurden auch kreisförmige Marker wie zum Beispiel WhyCon [75] oder Blobs [15] vorgeschlagen. Diese Familie von Markern kann jedoch keine ID im Marker encodieren.

Im Gegensatz zu visuellen Markern werden Thermalmarker durch Temperaturgradienten im Thermalbild sichtbar. Um den Herstellungsaufwand möglichst gering zu halten, wurden komplexe Marker wie April Tag [151] und Whycon [75] im Folgenden nicht betrachtet. Stattdessen wurden ArUco und Blobs näher analysiert, da diese aus den beiden Gruppen am wenigsten Rechenzeit benötigen und somit auch auf dem *Up-Core* Embedded-Computer echtzeitfähig sind [174].

Eine umfangreiche Evaluation der ArUco Marker Familie DICT_4X4_1000 hat ergeben, dass der Marker mit der ID 17 die beste Detektionsperformance aufweist. Hierbei wurden nur Marker evaluiert, die auch als Thermalmarker realisierbar sind. Experimentell wurde die Anzahl an erfolgreichen Detektionen in einem Datensatz, bei dem die Distanz und Blickwinkel variiert wurden, bestimmt. Marker ID 17 hat dabei die beste Performance erreicht und

wird daher hier verwendet. Anschließend wurden die maximalen Detektionsreichweiten in Bezug auf die Markergröße für diesen Marker bestimmt, siehe Tabelle 4.1.

Blobs sind verbundene Gruppen von Pixeln mit ähnlichen Eigenschaften [15]. Typische Eigenschaften sind die Größe, gleicher Intensitätswert, Rundheit und Konvexität. Insbesondere kreisförmige Blobs lassen sich sehr gut als Thermalmarker mittels einer Wärmequelle realisieren.

Kernbestandteil der Thermalmarker ist ein Peltierelement. Diese erzeugt bei angelegter Spannung eine Temperaturdifferenz zwischen der Ober- und Unterseite. Mit Hilfe eines passiven Hochleistungskühlkörpers, der an der Rückseite angebracht wurde, kann eine Temperaturdifferenz zur Umgebungstemperatur hergestellt werden. So erzeugt die Oberseite des Peltierelements einen deutlichen Kontrast im Thermalbild unabhängig von der eigentlichen Umgebungstemperatur.

Ein Peltierelement mit Kühlkörper ist in Abbildung 4.10 auf der linken Seite dargestellt. Die Oberseite kann mit dem Blob-Detektor sehr robust auch auf große Distanz detektiert werden. Detektionsreichweiten von mehr als 25 m konnten experimentell nachgewiesen werden.

Der ArUco Marker ID 17 wurde, wie auch das Kalibrierboard der Thermalkamera aus Unterabschnitt 3.1.3, mit einer beheizten Aluplatte sowie einer Holzschablone realisiert, wie in Abbildung 4.10 in der Mitte und rechts dargestellt. Durch den Kühlkörper auf der Rückseite des Peltierelements wird die Aluplatte relativ zur Umgebungstemperatur geheizt. Die Thermalmarker sind in Abbildung 4.11 in einem visuellen Bild (links) sowie im Thermalbild (rechts) gezeigt. Bereits geringe Temperaturgradienten von wenigen Grad sind ausreichend für einen deutlichen Kontrast.

Markergröße	Maximale Detektionsreichweite
10 cm	5 m
15 cm	8 m
20 cm	10 m
25 cm	12 m

Tabelle 4.1: Markergröße versus maximale Detektionsreichweite von ArUco Marker ID 17 nach Doer et al. [174].

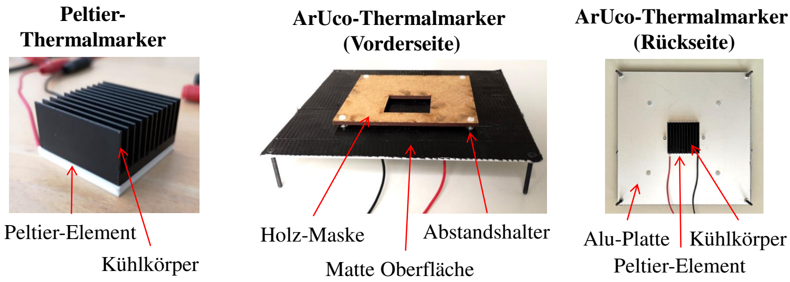


Abbildung 4.10: Entwickelte Peltier-Thermalmarker (links) sowie ArUco-Thermalmarker (Mitte und rechts).

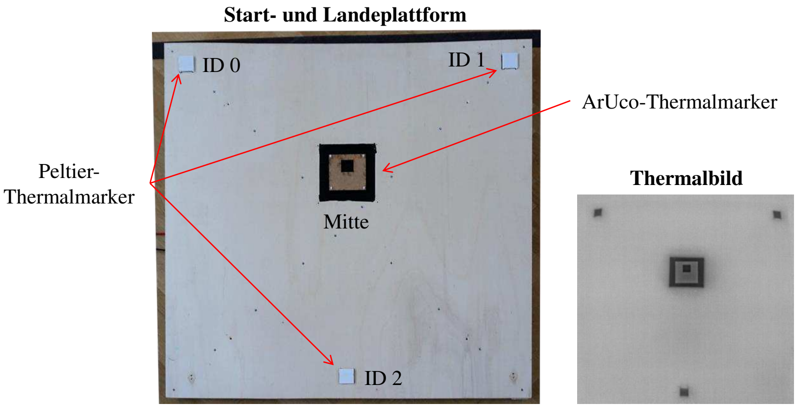


Abbildung 4.11: Entwickelte Start-und Landeplattform oben mit Thermalmarkertypen und IDs sowie einem korrespondierenden Thermalbild (unten rechts) nach Doer et al. [174].

4.4.2 Entwurf der Start-und Landeplattform

Ziel beim Entwurf der Start- und Landeplattform ist eine möglichst große Detektionsreichweite. Hier wurden mindestens 15 m gewählt [174]. Wie in Tabelle 4.1 hätte dies sehr große ArUco Marker (> 25 cm) zur Folge. Da die benötige Heizleistung proportional zum Flächeninhalt des Markers ist, sollte diese möglichst klein sein, um ein energieeffizientes System zu realisieren. Blobs können zwar keine ID integrieren, liefern aber auch bei sehr geringem

Energieaufwand ein hohe Detektionsreichweite. Ist jedoch nur ein Blob im Sichtbereich der Thermalkamera, zum Beispiel bei einer geringen Flughöhe, ist eine Assoziation des Blobs nicht möglich. Daher wurde beim Entwurf der Landeplattform ein hybrider Ansatz verfolgt, der einen kleinen ArUco Marker mit drei Peltier-Thermalmarkern kombiniert. Dabei wurde der ArUco Thermalmarker zentral angeordnet und bietet so bei geringen Flughöhen bis zur Landung die Möglichkeit einen Marker mit eindeutiger Assoziation im Sichtbereich zu haben. Die drei Peltier-Thermalmarker wurden für eine robuste Stützung des Gierwinkels maximal weit auseinander angeordnet. Abbildung 4.11 zeigt die Landeplattform im oberen Teil als visuelles Bild sowie unten rechts das korrespondierende Thermalbild.

Unter Einbeziehung des Sichtbereichs der Thermalkamera von 95°×73°, siehe Tabelle 3.1, kann der Sichtbereich der Start- und Landeplattform aufgestellt werden. Dies ist in Abbildung 4.12 skizziert. Dabei ist der Sichtbereich des ArUco Markers in hellblau und der Sichtbereich aller Blob-Marker in dunkelblau eingefärbt. Durch den unterschiedlichen Sichtbereich der beiden Achsen ergibt sich bei der Seitenansicht ein schmalerer Sichtbereich. Dennoch wird mit der entworfenen Plattform in 10 m eine Ellipse mit einer Nebendiagonalen von knapp 5 m erreicht. Dies ist größer als die Standardgenauigkeit von GPS in Open Sky Bedingungen [103]. Somit ist ein Anflug mit GPS möglich, da lediglich eine Genauigkeit von circa 5 m benötigt wird.

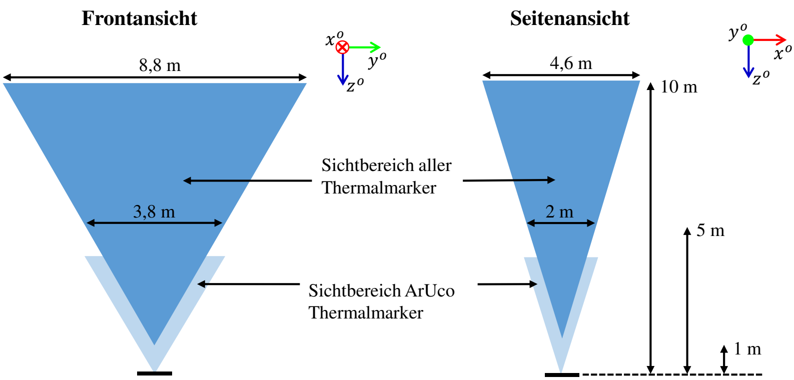


Abbildung 4.12: Sichtbereich der Start- und Landeplattform auf Basis der verwendeten Thermalkamera.

4.4.3 Detektion der Landeplattformmarker

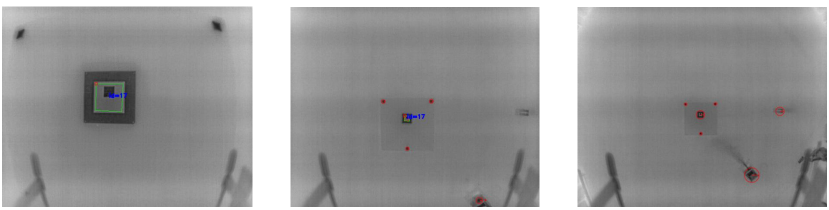
Wie im Abschnitt 4.4.2 motiviert, wurde ein Entwurf unter Minimierung des Energieverbrauchs angestrebt und daher ein möglichst kleiner ArUco Thermalmarker eingesetzt. In Abhängigkeit der Flughöhe ergeben sich dadurch die drei verschiedenen Modi der Landeplattformdetektion:

- Detektionsmodus 1 (geringe Flughöhen: $< 1\,\mathrm{m}$): Nur Detektion des ArUco Markers, da die Peltier-Thermalmarker aus dem Sichtbereich verschwinden.
- Detektionsmodus 2 (mittlere Flughöhen: 1 m - 4 m): Detektion des ArUco Markers und der drei Peltier-Thermalmarker.
- Detektionsmodus 3 (größere Flughöhen: >4 m): Detektion von Blobs für alle Thermalmarker, da auch der ArUco Marker nur robust als Blob detektierbar ist.

Die drei Detektionsmodi sind in Abbildung 4.13 veranschaulicht. Da die Höhe über der Landeplattform durch den nach unten gerichteten Radar gemessen wird, ist der Detektionsmodus eindeutig bestimmbar.

Ziel ist es anschließend, den ArUco Thermalmarker sowie die Peltier-Thermalmarker im Bild zu finden und ihrer ID, siehe Abbildung 4.11, zuzuordnen. Im Detektionsmodus 1 und 2 wird dafür zunächst der ArUco Detektor [116] angewendet. Anschließend erfolgt eine Detektion von Blobs-Features [15] was eine Liste an Features mit Bildkoordinaten $\boldsymbol{p}^{\mathrm{px}} = [u, v]$ liefert.

Höhe < 1 m | **Höhe 1 m – 4 m** | **Höhe > 4 m**



Modus: ArUco | Modus: ArUco & Blobs | Modus: Blobs

Abbildung 4.13: Veranschaulichung der drei Detektionsmodi in Abhängigkeit der Flughöhe

Zunächst erfolgt die Projektion der Bildkoordinaten $\boldsymbol{p}^{\mathrm{px}}$ in homogene Kamerakoordinaten $\tilde{\boldsymbol{p}}^{\mathrm{c}*}$

$$\tilde{\boldsymbol{p}}^{\mathrm{c}*} = \mathbf{K}^{-1} \begin{bmatrix} u & v & 1 \end{bmatrix}^T \tag{4.12}$$

mit der intrinsischen Kameramatrix $\mathbf{K}$. Anschließend werden Linsenverzerrungen entfernt, wobei das Fisheye-Linsenmodell [15] angewendet wird. Dies resultiert in den korrigierten homogenen Kamerakoordinaten $\boldsymbol{p}^{\mathrm{c}*}$. Mit der gemessenen Höhe über Landeplattform h_z, können die korrigierten homogenen Kamerakoordinaten in die Kartesische-Koordinaten $\boldsymbol{p}^c$ projiziert werden:

$$\boldsymbol{p}^c = h_z \, \boldsymbol{p}^{\mathrm{c}*}. \tag{4.13}$$

Schließlich erfolgt eine Transformation ins stabilisierte Koordinatensystem mit Hilfe der Roll- und Nickwinkel des Navigationsfilters aus der Navigationslösung:

$$\boldsymbol{p}_c^s = \mathbf{C}_b^s \left(\mathbf{C}_c^b \boldsymbol{p}^c - \boldsymbol{l}_{bc}^b \right). \tag{4.14}$$

Gleichungen (4.12) - (4.14) werden für alle detektieren Blobs angewendet. Anschließend erfolgt die Detektion der Landeplattform mittels eines Geometriemodells, dass in Abbildung 4.14 skizziert ist. Unter der Annahme, dass die Landeplattform zur Erdbeschleunigung ausgerichtet ist, also die z^o-Achse parallel zur Erdbeschleunigung ist, kann das Matching in 2D erfolgen. Ziel ist, den mittleren Marker sowie die drei äußeren ihrer ID zuzuordnen. Im Detektionsmodus 1 und 2 ist dies durch die Detektion des ArUco-Markers gelöst. Ein Matching kann dann durch die Evaluation der Distanzen d_0, d_1 und d_2 sowie der Winkel α_{01}, α_{02} und α_{12} durchgeführt werden. Hierzu wird der Fehler der Distanzen d_i und Winkel α_i für die projizierten Blobkoordinaten $\boldsymbol{p}^s$ zur realen Landeplattform bestimmt. Liegen die Abweichungen innerhalb empirisch ermittelter Grenzen (a für d_i und β für α_i) wird das Matching als erfolgreich betrachtet. Die Modellgrenzen a und β führen zu den in grün markierten Bereichen in Abbildung 4.14. Im Modus 3 dagegen, wird das zuvor beschriebene Verfahren für alle Blobfeatures iteriert. Bei mehr als einem validen Matching, wird das mit dem geringsten Distanzfehler ausgewählt. Auf

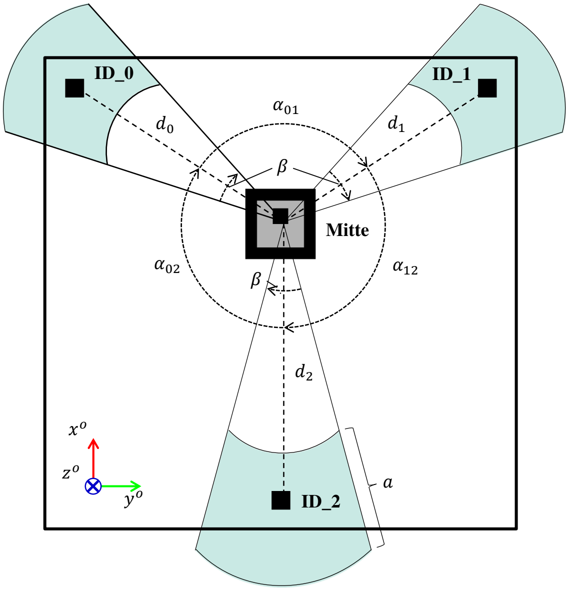


Abbildung 4.14: Geometriemodell der Landeplattform genutzt für das Matching

Basis des Landplattform Matchings, wird die relative Lokalisierung, wie im folgenden Abschnitt beschrieben, durchgeführt.

4.4.4 Navigationsstützung

Im Folgenden werden zwei Verfahren zur Lokalisierung präsentiert und anschließend evaluiert [174]:

- Posenschätzung 1: Lösung des Perspective-n-Point (PnP) Problem [123].
- Posenschätzung 2: Schätzung der Position und des Gierwinkels gestützt durch die globalen Roll- und Nickwinkel sowie der Radarhöhe.

Die Lösung des PnP-Problems zur Bild-basierten Posenschätzung ist sehr weit verbreitet und wird auch durch OpenCV [15] unterstützt. Dazu werden mindes-

tens drei Paare aus Bild-Welt-Punkt Korrespondenzen benötigt. Minimierung des Rückprojektionsfehlers liefert die Pose in sechs Freiheitsgraden. Die Genauigkeit der Posenschätzung skaliert dabei mit dem Abstand der Bildpunkte. Je größer diese, desto genauer die Posenschätzung, da die Detektion der Features in der Regel hier auf ganzzahlige Pixel genau ist. Besonders schwierig für dieses Verfahren sind daher Szenen, in denen alle Bildpunkte sehr nahe beieinanderliegen. Da die Landeplattform nur 1×1 m groß ist, ergeben sich schwierige Szenen bereits ab circa 3 m [174]. Daher wurde mit der Posenschätzung 2 ein weiteres Verfahren entwickelt [174].

Die Posenschätzung 2 nutzt zusätzlich zu den Bildpunkten der Landeplattform auch die Höhe über dieser mit Hilfe des Radaraltimeters. Zusätzlich wird angenommen, dass der Roll- und Nickwinkel des VTOLs bereits mit ausreichender Güte durch die Navigationslösung verfügbar ist. Somit reduziert sich die Posenschätzung auf die Position $\boldsymbol{p}_b^o$ und dem Gierwinkel ψ_{ob}. Es wird ein ähnlicher Ansatz wie beim Radar-basierten System, siehe Unterabschnitt 4.3.4, verfolgt. Zunächst wird der Gierwinkel berechnet. Unter der Annahme, dass die Landeplattform zu den Achsen des Navigationskoordinatensystems ausgerichtet ist, wie in Abbildung 4.14 dargestellt, können die einzelnen Gierwinkelmessungen ψ_i aufgestellt werden:

$$\psi_{c0} = \operatorname{atan2}(p^s_{\mathrm{ID0},y} - p^s_{\mathrm{Mitte},y}, p^s_{\mathrm{ID0},x} - p^s_{\mathrm{Mitte},x}), \tag{4.15}$$

$$\psi_{c1} = \operatorname{atan2}(p^s_{\mathrm{ID1},y} - p^s_{\mathrm{Mitte},y}, p^s_{\mathrm{ID1},x} - p^s_{\mathrm{Mitte},x}), \tag{4.16}$$

$$\psi_{c2} = \operatorname{atan2}(p^s_{\mathrm{ID2},y} - p^s_{\mathrm{Mitte},y}, p^s_{\mathrm{ID2},x} - p^s_{\mathrm{Mitte},x}), \tag{4.17}$$

wobei werden die Bildpunkte der IDs $0-2$ sowie der mittlere Marker mit (4.14) ins stabilisierte Koordinatensystem transformiert wurden. Mittelung über alle Messungen ergibt die Gierwinkelmessung $\psi_{ob,m}$:

$$\psi^o_{ob,m} = \frac{1}{3} \sum_{i=0}^{2} \psi_{ci}. \tag{4.18}$$

Somit kann die Messung der globalen Pose für jede ID $\{0, 1, 2, \mathrm{Mitte}\}$ unter Nutzung der globalen Markerposition $\boldsymbol{p}^o_{\{0,1,2,\mathrm{M}\}}$ aufgestellt werden

$$\boldsymbol{p}^o_{ob,\{0,1,2,\mathrm{Mitte}\}} = \mathbf{C}^o_b \left(\mathbf{C}^b_c \boldsymbol{p}^c_{\{0,1,2,\mathrm{M}\}} - \boldsymbol{l}^b_{bc} \right) - \boldsymbol{p}^o_{\{0,1,2,\mathrm{M}\}}, \tag{4.19}$$

wobei die finale Messung der globalen Pose $\boldsymbol{p}^{o}_{ob,m}$ wieder über Mittelung aller Messungen erfolgt:

$$\boldsymbol{p}^{o}_{b,m} = \frac{1}{4} \sum_{i \in \{0,1,2,\mathrm{M}\}} \boldsymbol{p}^{o}_{ob,i}. \tag{4.20}$$

Analog zum Radar-basierten System, werden die Messungen der Position $\boldsymbol{p}^{o}_{ob,m}$ und des Gierwinkel $\psi_{ob,m}$ mit den Messdaten der IMU in einem Kalman-Filter fusioniert. Hierbei wird ein Mahalanobis Test, wie im Anhang A.1 erläutert, durchgeführt, um Outlier herauszufiltern.

4.4.5 Ergebnisse

Im Folgenden werden die Ergebnisse erst mittels Simulation und anschließend mit Realfügen mit Motion-Capture sowie in schlechten Sichtbedingungen präsentiert.

Simulation

Eine Verifikation des Ansatzes wurde zunächst in Simulation durchgeführt. Dabei wurde eine realitätsnahe Simulation mit Gazebo [73], wie in Unterabschnitt 3.2.2 beschrieben, durchgeführt. Eine Visualisierung der Simulation ist in Abbildung 4.15 links dargestellt. Ein Vergleich zwischen dem simulierten und einem realen Thermalbild ist in Abbildung 4.15 in der Mitte und rechts gezeigt. Hierbei zeigt die Simulation eine realistische Darstellung der Landeplattform. Während die maximale Detektionsreichweite in der Simulation bei circa 10 m liegt, wird mehr als 15 m beim realen System erreicht. Da die Peltierelemente als Leuchtquelle im Thermalbild erscheinen, wird in der Simulation lediglich eine passive Struktur simuliert. Die Sensordaten der IMU sowie das Flugverhalten wurden realistisch parametrisiert. Zwei Simulationssetups analysieren dabei verschiedene Aspekte der Navigationsstützung. Der Treppenflug evaluiert die Genauigkeit in Abhängigkeit der Höhe und der Kreisflug die Performance bei Variation der horizontalen Position nahe der Detektionsgrenze.

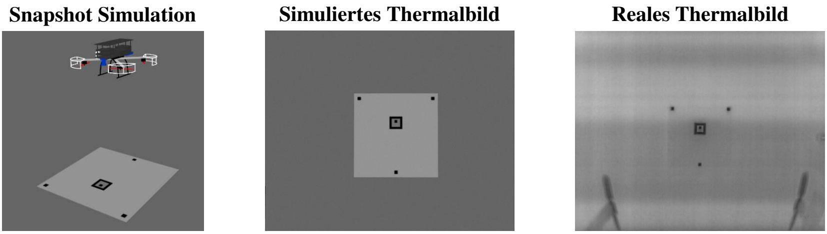


Abbildung 4.15: Simulation (links) mit simulierten Thermalbild (Mitte) und zum Vergleich ein reales Thermalbild (rechts) basierend auf Doer et al. [174].

Beim Treppenflug wird ein stufenförmiges Höhenprofil abgeflogen, das den Bereich 1 – 9 m abdeckt. Alle drei Sekunden wird dabei die nächste Stufe eingestellt, wie in Abbildung 4.16 oben dargestellt ist. Die beiden Posenschätzungsansätze erzielen unterschiedliche Positionsfehler, wie Abbildung 4.16 in der Mitte zu erkennen ist. Insbesondere bei Höhen oberhalb 4 m zeigen sich beim Ansatz 1 Positionsfehler etwa um den Faktor 10 größer als beim Ansatz 2. Die Fehler beider Ansätze skalieren mit der Höhe, daher zeigen sich im Bereich um 30 s die größten Positionsfehler. Eine Analyse der Gierwinkelfehler ist im unteren Bereich von Abbildung 4.16 dargestellt. Hierbei erzielen beide Ansätze eine ähnliche Genauigkeit. Unter 1 m Höhe (im Bereich von 6 s oder 60 s wird nur der ArUco Marker zur Lokalisierung genutzt, da die Peltiermarker nicht vollständig im Sichtbereich sind. Da dann die Bildpunkte sehr nahe beieinanderliegen, ergibt sich eine schlechtere Schätzung des Gierwinkels.

Das zweite Simulationsszenario testet das Gesamtsystem nahe der Detektionsgrenze. Hierzu wird ein vollständiger Flug gemäß dem Szenario von Abbildung 1.1 simuliert. Es folgt zunächst ein Start bis in eine Höhe von 8 m, anschließend wird ein kreisförmiger Pfad mit Radius 2 m mittig oberhalb der Landeplattform abgeflogen und schließlich gelandet. Die Draufsicht auf den Flugpfad ist in Abbildung 4.17 oben abgebildet. Während des kreisförmigen Fluges wird der Gierwinkel so eingeregelt, dass die Flugplattform immer in Bewegungsrichtung zeigt. Dadurch wird der volle Gierwinkelbereich getestet. Während des gesamten Fluges konnte die Pose erfolgreich mit dem Ansatz 2 geschätzt werden, wie in Abbildung 4.17 unten gezeigt ist. Dabei wird eine mittlere Positionsfehlernorm von 0,13 m sowie ein mittlerer Gierwinkelfehlerbetrag von 0,35° erreicht.

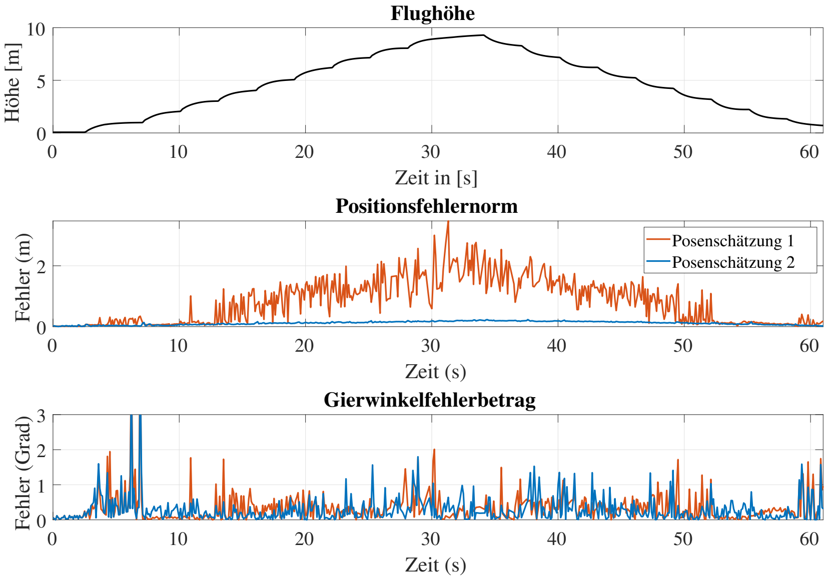


Abbildung 4.16: Simulationsszenario mit treppenförmigem Höhenprofil nach Doer et al. [174]

Hardwareaufbau

Die folgenden Flugexperimente wurden mit der leicht modifizierten Flugplattform aus Unterabschnitt 4.3.5, siehe Abbildung 4.18 links, durchgeführt. Dabei wurde die Thermalkamera nach unten ausgerichtet montiert. Es wird wieder das *Up-Core* als Embedded-Computer eingesetzt.

Experimentelle Ergebnisse mit Motion-Capture

Analog zur Evaluation des Radar-basierten Systems wurde ein Testflug mit Motion-Capture Referenzdaten im Bereich $0,5 - 2$ m oberhalb der Landeplattform ausgeführt. Mit Hilfe der Motion-Capture Referenz können die Schätzfehler der Pose exakt ausgewertet werden wie in Abbildung 4.19 analysiert. Hierbei wurde das Höhenprofil, siehe Abbildung 4.19 oben, abgeflogen. Zwischen $22 - 40$ s wurde dabei die gleiche Höhe gehalten und die horizontale

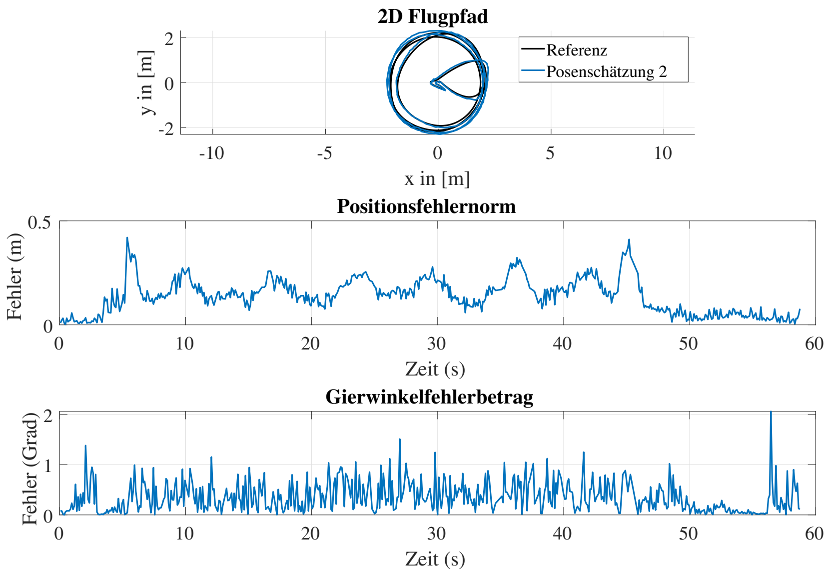


Abbildung 4.17: Simulationsszenario mit kreisförmigen Flug in 8 m Höhe nach Doer et al. [174]

Position variiert. Analog zur Simulation wird auch hier die Position und der Gierwinkelfehler der beiden Posenschätzungsansätze evaluiert.

Beide Ansätze erzielen eine hohe Genauigkeit bei der Positionsschätzung. So erreicht die Posenschätzung 1 eine mittlere Positionsfehlernorm von 0,15 m und der Ansatz 2 von 0,11 m. Unterschiede zeigen sich vor allem bei der Gierwinkelschätzung. Ansatz 1 erzielt dabei einen mittleren Gierwinkelfehler von 3,2° während Ansatz 2 mit 0,5° einen deutlich geringeren Fehler erreicht.

Experimentelle Ergebnisse Sporthalle

Eine Evaluation des Gesamtsystems wurde mit 10 autonomen Testflügen in einer Sporthalle durchgeführt. Dabei wurde die gleiche Flugführung wie auch bei den Testflügen des Radar-basierten Systems genutzt. Die Hover Höhe wurde dabei zwischen 2–5 m variiert und ist durch die Höhe der Hallendecke limitiert.

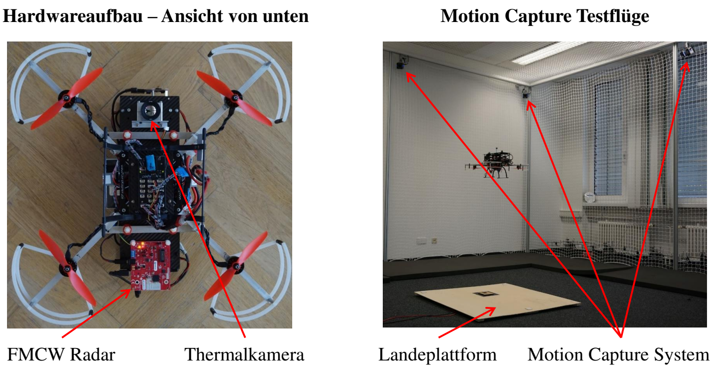


Abbildung 4.18: Modifiziertes Indoorflugsystem (links) und Evaluation mit Motion-Capture (rechts) basierend auf Doer et al. [174].

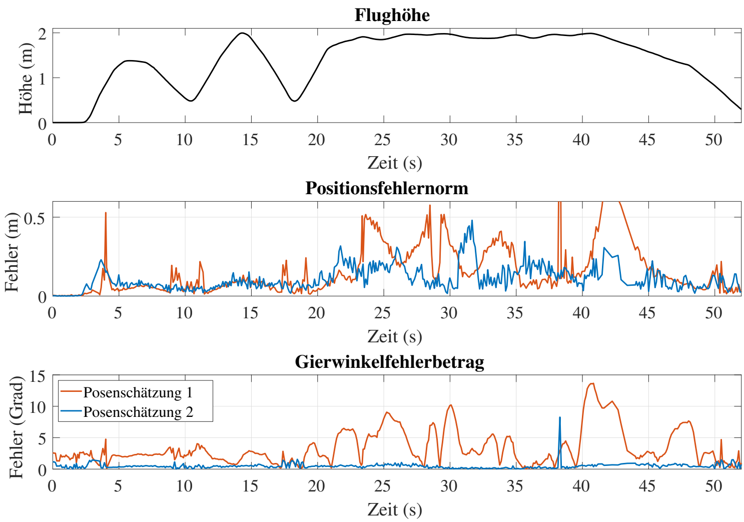


Abbildung 4.19: Evaluation mit Motion-Capture

Da die Posenschätzung 2 eine deutlich bessere und robustere Posenschätzung liefert, wurde diese eingesetzt. Einer der Testflüge in guten Sichtbedingungen ist in Abbildung 4.20 auf der linken sowie bei schlechten Sichtbedingungen auf der rechten Seite dargestellt. Die korrespondierenden Thermalbilder sind unten abgebildet, wobei das jeweils linke Bild das rohe Thermalbild und das rechte das Ergebnis der Landeplattformdetektion zeigt. Die Detektion ist sehr robust und auch die Hitzequelle der Nebelmaschine beeinflusst die Detektion nicht. Alle 10 Testflüge waren erfolgreich, was die Robustheit des Systems unter Beweis stellt.

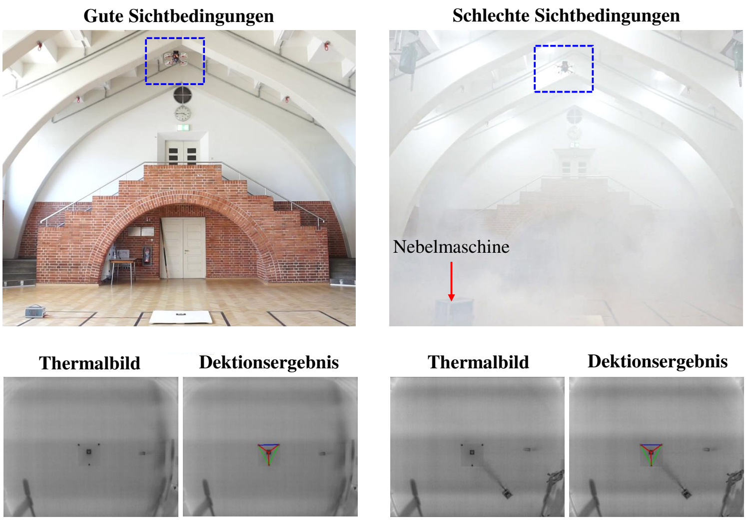


Abbildung 4.20: Testflüge bei guten (links) und schlechten Sichtbedingungen (rechts) mit dem Input Thermalbild (jeweils links) sowie dem Ergebnis der Landeplattformdetektion (rechts).

4.5 Fazit

In diesem Kapitel wurden zwei Systeme zur Navigationsstützung für präzise Start- und Landemanövern für VTOLs präsentiert. Eine Übersicht beider Systeme ist in Abbildung 4.21 gegeben und Tabelle 4.2 vergleicht die wichtigsten Kriterien.

Beide Systeme weisen die gleiche Größe der Start- und Landeplattform auf und wurden mit Testflügen erfolgreich demonstriert. Das Radar-basierte System kommt mit einer rein passiven Landeplattform aus, weist nur eine geringe CPU Nutzung auf und kann mit günstiger Sensorik umgesetzt werden. Die Detektionsreichweite sowie die Genauigkeit der Posenschätzung ist jedoch schlechter. Das Radar-Thermalkamera-basierte System dagegen erzielt eine deutlich bessere Performance bei der Detektionsreichweite und Posenschätzung. Es benötigt dafür jedoch eine aktive Landeplattform, weist eine höhere CPU Nutzung auf und ist durch die Thermalkamera deutlich teurer.

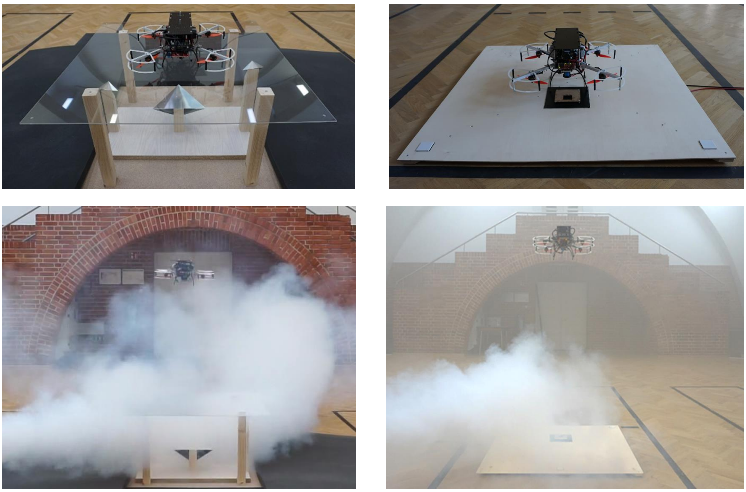


Abbildung 4.21: Übersicht des Radar-basierten (links) und des Thermalkamera-basierten (rechts) Systems zur Start- und Landestützung.

Kriterium	Radar	Thermalkamera und Radar
Schlechte Sichtbedingungen	ja	ja
Größe Landeplattform	1 m × 1 m	1 m × 1 m
Art der Landeplattform	passiv	aktiv: $P = 10\,\mathrm{W}$
Genauigkeit Positionschätzung	gut	sehr gut
Genauigkeit Gierwinkelschätzung	mittel	sehr gut
Maximale Detektionsreichweite	ca. 5 m	ca. 15 m
Echtzeitfähig auf *Up-Core*	ja	ja
CPU Nutzung	gering	hoch
Maximale Detektionsrate	20 Hz	10 Hz
Kosten Sensorik	günstig	teuer
Gewicht Sensorik	50 g	50 g + 80 g

Tabelle 4.2: Vergleich der Start- und Landesysteme: Radar-basiert versus Thermalkamera-basiertes System.

Beide Systeme ermöglichen autonome Start- und Landemanöver der motivierten Szenarien nach Abbildung 1.1 auch in schlechten Sichtbedingungen, wie umfangreich demonstriert wurde. Somit wird das Ziel, Verfahren für Start- und Landemanöver zu entwickeln, die auch bei schlechten Sichtbedingungen autonome Flüge ermöglichen, erreicht.

5 Radar-inertiale Navigation

Robuste und akkurate Navigation auch in schlechten Sichtbedingungen ist eine Voraussetzung autonomer Flugmissionen, beispielsweise für Such- und Rettungsmissionen aber für Anwendungen wie die Lieferung und Navigation in Umgebungen ohne GNSS-Empfang. Radar wird auf Grund der Robustheit gegenüber Umwelteinflüssen wie zum Beispiel Regen, Schnee, Nebel oder Dunkelheit bereits sehr lange im Luftfahrtbereich eingesetzt [91]. Dieses Kapitel stellt die Radar-inertiale Navigation vor und stellt eine Zusammenfassung der Veröffentlichungen Doer et al. [178, 179, 181, 182] dar. Abbildung 5.1 zeigt den Systemaufbau mit den wichtigsten Komponenten. Mit den Messdaten der IMU wird die Propagation durchgeführt sowie das Barometer-Update. Die Radar-Trigger-Signale markieren den Start eines Radarscans und lösen die Zustandserweiterung aus. Die Scans der Radarsensoren werden zunächst von der Eigengeschwindigkeitsschätzung verarbeitet, woraus sich die Radar-Eigengeschwindigkeit $\boldsymbol{v}^r_{\text{radar}}$ mit zugehöriger Kovarianz $\mathbf{P}_{\text{radar}}$ ergibt. Auf Basis der Inlier der Eigengeschwindigkeitsschätzung, also der statischen Objekte des Radarscans, wird mit Hilfe der Gierwinkelschätzung die Gierwinkelmessung ψ^o_{radar} generiert. Die Fusion wird mit einem Error-State-Kalman-Filter, siehe Abschnitt 2.2.2 im Grundlagen-Kapitel, implementiert. Im Folgenden wird zunächst die Radarscanverarbeitung (gelbe Blöcke) näher beschrieben und anschließend das Navigationsfilter (blau gestrichelter Block) präsentiert. Es erfolgt die Evaluation der Radarscanverarbeitung sowie des Gesamtsystems mit Hilfe einer umfassenden Sammlung an Realdatensätzen sowie der Demonstration der Online-Navigation von UAS.

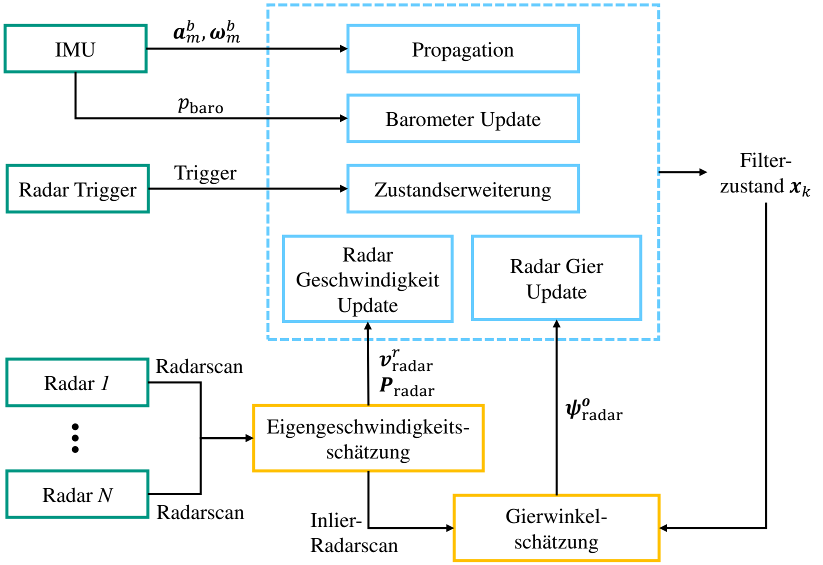


Abbildung 5.1: Blockdiagramm des Radar-inertialen Navigationssystems mit Sensordaten (grün), dem Navigationsfilter (blau gestrichelt) mit dessen wichtigsten Komponenten (blau), und der Radarscanverarbeitung (gelb).

5.1 Verwandte Arbeiten

Navigation ohne GNSS-Empfang

Die Navigation in Umgebungen ohne GNSS-Empfang wurde bereits sehr intensiv erforscht. Ein sehr bekannter Ansatz ist die Visuelle-inertiale Odometrie (VIO), die beispielsweise auch die Navigation von UAS ermöglicht [12, 33, 110]. Eine Analyse der Open-Source VIO-Pipelines von Delmerico et al. [33] zeigt, dass von den untersuchten Ansätzen VINS [110] die beste Genauigkeit erreicht. VIO-Ansätze sind jedoch auf eine gut beleuchtete Szene angewiesen und können, wie im Abschnitt 3.1.1 experimentell gezeigt, nicht bei schlechten Sichtbedingungen eingesetzt werden. VINS wird jedoch im folgenden Kapitel zum Vergleich zur Radar-inertialen Navigation in guten Sichtbedingungen mit

evaluiert. So kann ein Vergleich zu einem etablierten VIO-Verfahren durchgeführt werden.

Navigation in dunklen Umgebungen kann beispielsweise durch den Einsatz von Lidar ermöglicht werden. So fusioniert [158] die Daten eines Laserscanners mit Inertialsensorik, was zu einer sehr genauen Bewegungsschätzung und Kartierung der Umgebung führt. Wie jedoch in Abbildung 3.1 gezeigt, können Laserscanner in schlechten Bedingungen, wie zum Beispiel Nebel, nicht eingesetzt werden.

Inspiriert von VIO, wurde thermale-inertiale Odometrie (TIO) vorgeschlagen [32, 69, 72, 126, 160]. Da Thermalbilder nicht von schlechten Sichtbedingungen wie Dunkelheit oder Nebel beeinflusst werden, stellt TIO einen vielversprechenden Navigationsansatz dar. Herausfordernd sind jedoch die sogenannten Non-Uniformity-Corrections (NUCs) der Thermalkameras sowie Szenen mit geringen Temperaturgradienten. NUCs sind regelmäßige Selbstkalibierroutinen der Thermalkamera, um beispielsweise ein Aufwärmen des Sensors zu korrigieren. Während einer NUC (bis zu 0,75 s) stellen Thermalkameras keine Bilder bereit. Zusätzlich weisen insbesondere Szenen innerhalb von Gebäuden häufig geringe Temperaturgradienten auf. Dies führt zu Bildern mit wenig markanten Strukturen. Beide Gegebenheiten führen dazu, dass reine TIO-Ansätze nicht für ein robustes Navigationssystem genutzt werden können.

Radar-basierte Navigation

Die Verwendung von Radar zeigt in Szenarien schlechter Sichtbedingungen den vielversprechendsten Ansatz, wie in Abbildung 3.1 gezeigt ist. Zusätzlich ist Radar sehr robust gegenüber Witterungsbedingungen [91] und wird daher in Automotive Anwendungen bereits seit langem für Fahrerassistenzsysteme eingesetzt [34]. Auch beim autonomen Fahren wird Radar eine entscheidende Rolle zugesprochen [34]. Eigenbewegungsschätzung mit Hilfe von Radar kann in drei Gruppen eingeteilt werden: Scanmatching-Ansätze [4–6, 20, 113, 131], instantane-Ansätze [67, 68] und hybride Ansätze [9,95]. Scanmatching-Ansätze sind sehr etabliert und wurden bereits zur Bewegungsschätzung mittels Laserscannern intensiv erforscht. Auf dieser Basis wurden auch Scanmatching-Verfahren mit Radar entwickelt [4–6,20,113,131]. Dabei werden Merkmalspunkte über mehrere Scans hinweg verfolgt und damit

die relative Posenänderung bestimmt [6,20,113]. Alternativ kann der aktuelle Scan auch gegen eine bestehende oder online erstellte Karte gematcht werden [131]. Nachteilig bei Verfahren basierend auf Scanmatching ist die reduzierte Performance bei sehr dynamischer Bewegung und sich stark ändernden Szenen. Ein Verfolgen von Features über mehrere Scans hinweg ist dann nicht mehr möglich und die Bewegungsschätzung verliert stark an Genauigkeit. Daher wird im Rahmen dieser Arbeit ein instantaner Ansatz verfolgt.

Instantane Ansätze ermöglichen die Schätzung der Radar-Eigengeschwindigkeit auf Basis eines einzelnen Radarscans. Dabei wird die Richtung sowie die Dopplergeschwindigkeit zu jedem detektierten Objekt genutzt, um die Eigengeschwindigkeit zu berechnen. Ein Verfolgen von markanten Punkten über mehrere Scans wird nicht benötigt. Kellner et al. [67] präsentierten dies zunächst für die zweidimensionale Geschwindigkeitsschätzung für automotive Anwendungen. In [68] wird eine Erweiterung für den Fall mehrerer Radarsensoren beschrieben. Durch die Verwendung von Radarsensoren, die vorne und hinten am Fahrzeug angebracht sind, kann neben der zweidimensionalen Eigengeschwindigkeit auch die Gierrate geschätzt werden. Hierfür sind mindestens zwei Radarsensoren notwendig und ein möglichst großer Hebelarm, also Abstand zwischen den Radarsensoren, ist wichtig für eine ausreichend genaue Schätzung der Gierrate [67]. Im Rahmen dieser Arbeit liegt der Fokus jedoch auf 3D-Navigation weshalb eine Stützung mit zweidimensionaler Eigengeschwindigkeit und Gierrate nicht ausreichend ist. Zudem sind Hebelarme, die größer als 1 m sind, auf kleinen Fluggeräten nicht umsetzbar. Die Schätzung der dreidimensionalen Geschwindigkeit aus Dopplermessungen wurde bereits in der Meteorologie zur Sturmmessung eingesetzt [10,22]. Mit Hilfe der Dopplermessungen von benachbarten Sturmzellen kann so die dreidimensionale Geschwindigkeit der Hauptsturmzelle geschätzt werden [10]. Dabei wird ein lineare Least-Squares Ansatz eingesetzt. In dieser Arbeit wird darauf aufbauend die Eigenbewegungsschätzung durchgeführt.

Weiterhin kann Radar-basierte Bewegungsschätzung bezüglich der eingesetzten Radarsensoren eingeteilt werden: Einstrahl-Radarsensoren [111, 112], Scanning-Radarsensoren [4, 20, 107] und Single-Chip-mmWave-Radarsensoren [6, 76, 86]. Quist et al. [111, 112] nutzen einen Einstrahl-FMCW-Radarsensor für die Navigation eines Luftfahrzeugs. Eine Winkelmessung ist mit solch einem Radarsensor nicht möglich. Durch horizontale Bewegung eines Luftfahrzeugs über dem Boden kann jedoch die Geschwindigkeit aus den

akkumulierten Messungen bestimmt werden [112]. Dies ist jedoch nur bei rein horizontaler Bewegung möglich und ist daher für Szenarien in Innenräumen oder mit vertikaler Bewegung nicht einsetzbar. Scanning-Radarsensoren sind analog zu Laserscannern aufgebaut. So wird ein Radarsensor mit sehr engem Sichtbereich horizontal rotiert, um einen Scan der Umgebung zu erstellen. Da diese Art von Sensor recht groß ist, werden diese nur für automotive Anwendungen eingesetzt [4, 20, 107]. Single-Chip-mmWave-Radarsensoren sind am vielversprechendsten bezüglich der in dieser Arbeit betrachteten Szenarien. Mit Hilfe von mehreren Sende- und Empfangsantennen können 3D-Scans der Umgebung mit einem statischen Sensor erzeugt werden. Diese Sensoren sind erst seit wenigen Jahren verfügbar und daher Gegenstand aktueller Forschung [6, 76, 86].

Radar-inertiale Navigation

Die Fusion mit Inertialsensorik hat sich als sehr gewinnbringend herausgestellt und nutzt die komplementären Eigenschaften der verwendeten Sensoren aus. So sind die Messdaten einer IMU mit einer hohen Rate verfügbar ($>$ 200 Hz), unabhängig von der Umgebung und Infrastruktur. Auf Grund sich ändernder Offsets der Beschleunigungs- und Drehratensensoren insbesondere bei den eingesetzten MEMS-IMU, akkumuliert sich sehr schnell ein Positionsfehler. Radar-basierte Bewegungsschätzung biete dagegen zwar eine niedrigere Messrate (20 Hz) und benötigt Struktur in der Umgebung, die Messungen sind allerdings frei von Offsets. Die Fusion von Radar und Inertialsensorik, sogenannte Radar-inertiale Odometrie (RIO), ermöglicht ein Navigationssystem mit hoher Rate und Genauigkeit. Dies ermöglicht die Schätzung der 3D-Pose mit sechs Freiheitsgraden [6, 76, 86]. Kramer et al. [76] nutzten einen optimierungsbasierten Ansatz über ein gleitendes Fenster. Der Fokus liegt jedoch auf der Schätzung der Geschwindigkeit und nicht auf der Schätzung der vollen Navigationslösung. Lu et al. [86] verwenden Deep Neural Nets (DNN) für das Scanmatching und zur Zustandsschätzung. Almalioglu et al. [6] setzen Recurrent Neural Nets (RNN) für das Scanmatching und einen Kalman-Filter für die Zustandsschätzung ein.

Wie in Abschnitt 1.3 motiviert, soll in dieser Arbeit die online Navigation auf Rechenplattformen mit sehr begrenzten Ressourcen ermöglicht werden. Dies schließt die Ansätze [6,86] aus, da die neuronalen Netze zusätzliche Ressourcen

benötigen. Alle drei Verfahren [6, 76, 86] benötigen eine exakte Kalibrierung des Radarsensors bezüglich der Translation und Rotation zum Bodyframe (definiert durch die IMU). Dies erfordert eine händische Kalibrierung des Setups. Des Weiteren verwenden [6, 76, 86] nur einen einzelnen Radarsensor. Je nach Umgebung und Charakteristik ist die Verwendung mehrerer Radarsensoren jedoch sehr vorteilhaft in Bezug auf Robustheit und Genauigkeit.

Fazit

Keine der in der Literatur beschriebenen Ansätze erfüllt alle geforderten Anforderungen gemäß der definierten Ziele aus Abschnitt 1.3. So können visuelle Ansätze nicht in schlechten Sichtbedingungen eingesetzt werden. Auf Grund der Größe und Qualität des Radarscans eignen sich nur Single-Chip-Radarsensoren für den betrachteten Anwendungsfall der Navigation von Fluggeräten. Für eine robuste Bewegungsschätzung auch bei hoher Dynamik wird der Ansatz der instantanen Geschwindigkeitsschätzung in dieser Arbeit weiterentwickelt. Es wird im Folgenden ein neuartiger Ansatz zur Radar-inertialen Navigation beschrieben. Dabei wird der Stand der Technik von Radar-basierten Stützinformation weiterentwickelt um:

- Ein robustes RANSAC-basiertes instantanes Schätzverfahren, um die 3D Radar-Eigengeschwindigkeit mit einem einzelnen Radarscan exakt zu bestimmen [178].
- Die Generierung von Radar-Gierwinkelmessungen basierend auf Manhattan-Welt-Annahmen, die nach Initialisierung mittels eines einzelnen Radarscans zur Verfügung stehen [181].

Die Radar-basierten Stützinformationen werden dann mittels eines neu entwickelten Navigationsfilters mit Inertial- und Barometersensordaten fusioniert, das die folgenden Eigenschaften aufweist [178, 179, 181, 182]:

- Die Fusion von Inertialsensordaten und Radar-basierten Stützinformationen erfolgt mittels eines Error-State-Kalman-Filters.
- Die robuste Verarbeitung der latenzbehafteten Radar-basierten Stützinformationen wird mit Hilfe von Stochastic Cloning gelöst.

- Die online Kalibrierung der Transformation zwischen dem Koordinatensystem der IMU und des Radarsensors als Teil des Filterzustandes durchgeführt.
- Der Ansatz ist generalisiert für eine frei konfigurierbare Anzahl an Radarsensoren.
- Der entwickelte Ansatz ermöglicht die robuste und exakte Navigation in Innenräumen als auch außerhalb von Gebäuden in urbanen Gebieten in guten und schlechten Sichtbedingungen.
- Die Demonstration für die online Navigation eines UAS mit einem Embedded-Computer auch bei schlechten Sichtbedingungen beweist die praktische Anwendbarkeit des entwickelten Systems.

Umfassende Evaluationen der Teilkomponenten als auch des Gesamtsystems in Simulation und Realdatensätzen werden im Laufe des Kapitels präsentiert. Die Demonstration der online Navigation erfolgt mit zwei autonomen Flügen unter guten sowie schlechten Sichtbedingungen.

5.2 Radar-Eigengeschwindigkeitsschätzung

Dieser Abschnitt beschreibt die Radar-Eigengeschwindigkeitsschätzung. Mit Hilfe von Richtungsinformationen und Doppler-Geschwindigkeitsmessungen aus dem Radarscan kann diese mit einem einzelnen Radarscan berechnet werden. Hierfür wird zunächst das Messmodell erläutert und Lösungsmöglichkeiten mit Least-Squares-Methoden diskutiert. Das Messmodell wird dann mit Hilfe eines robusten RANSAC-basierten Verfahrens erweitert und anschließend um eine nachfolgende Optimierung mit einer Orthogonal Distance Regression (ODR) verbessert. Die hier aufgeführte Radar-Eigengeschwindigkeitsschätzung wurde zuvor in den Publikationen Doer et al. [178, 180] veröffentlicht.

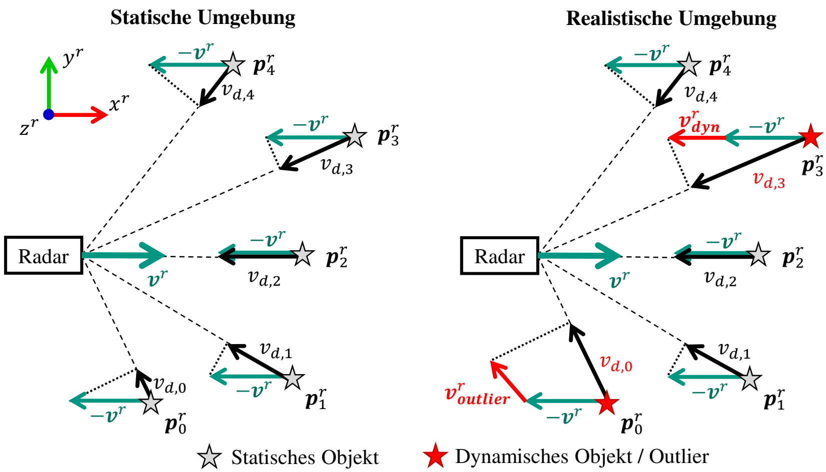


Abbildung 5.2: Radar-Eigengeschwindkeitsschätzung in einer statischen Szene (links) und einer realistischen Szene mit dynamischen Objekten (rechts).

5.2.1 Messmodell

Abbildung 5.2 zeigt auf der linken Seite einen Radarscan bestehend aus fünf statischen Objekten mit den Positionen $\boldsymbol{p}_i^r$. Der Radarsensor bewegt sich von links nach rechts mit der Geschwindigkeit $\boldsymbol{v}^r$. Dadurch ergeben sich die Doppler-Geschwindigkeiten $v_{d,i}$, durch schwarze Pfeile gekennzeichnet.

Es folgt für die Dopplergeschwindigkeit $v_{d,i}$ des i-ten Scanpunktes bei einer Radar-Eigengeschwindigkeit $\boldsymbol{v}^r$:

$$-v_{d,i}^r = \frac{\boldsymbol{p}_i^r}{||\boldsymbol{p}_i^r||} \circ \boldsymbol{v}^r = \boldsymbol{r}_i^r \cdot \boldsymbol{v}^r = r_{x,i}^r v_x^r + r_{y,i}^r v_y^r + r_{z,i}^r v_z^r. \tag{5.1}$$

Die Dopplergeschwindigkeit im Radar wird relativ zum Objekt gemessen. Da die Radar-Eigengeschwindigkeit geschätzt werden soll, muss diese negiert

werden. Ist ein Radarscan mit N Objekten gegeben, ergibt (5.1) in Matrixform das Messmodell:

$$\underbrace{\begin{bmatrix} -v^r_{\mathrm{d},1} \\ -v^r_{\mathrm{d},2} \\ \vdots \\ -v^r_{\mathrm{d},N} \end{bmatrix} = \begin{bmatrix} r_{x,1} & r_{y,1} & r_{z,1} \\ r_{x,2} & r_{y,2} & r_{z,2} \\ \vdots & \vdots & \vdots \\ r_{x,N} & r_{y,N} & r_{z,N} \end{bmatrix} \begin{bmatrix} v^r_x \\ v^r_y \\ v^r_z \end{bmatrix}}_{\boldsymbol{y}^r \,=\, \mathbf{H}\, \boldsymbol{v}^r}. \tag{5.2}$$

5.2.2 Lösung mit Least-Squares

Bei 5.2 handelt es sich um ein lineares Schätzproblem. Unter der Annahme, dass es sich bei dem Messrauschen der Dopplergeschwindigkeiten v^r_d um Gaußsches Rauschen handelt, kann die optimale Lösung mittels Least-Squares [102] berechnet werden. Dies erfolgt durch die Moore-Penrose-Pseudoinverse mittels [102]:

$$\hat{\boldsymbol{v}}^r = \left(\mathbf{H}^T\mathbf{H}\right)^{-1}\mathbf{H}^T\boldsymbol{y}^r. \tag{5.3}$$

Neben der Radar-Eigengeschwindigkeitsschätzung $\hat{\boldsymbol{v}}^r$ ist auch deren Unsicherheit für die nachfolgende Datenfusion wichtig. Nach [51] kann die Kovarianzmatrix $\mathbf{P}_r \in \mathbb{R}_{3\times 3}$ durch

$$\mathbf{P}_r = \frac{(\boldsymbol{e}\boldsymbol{e}^T)(\mathbf{H}^T\mathbf{H})^{-1}}{\dim(\boldsymbol{y}^r) - 3}, \tag{5.4}$$

$$\boldsymbol{e} = \mathbf{H}\boldsymbol{v}^r - \boldsymbol{y}^r \tag{5.5}$$

abgeschätzt werden. Mit Hilfe des Messmodells (5.2) kann somit die Radar-Eigengeschwindigkeit $\hat{\boldsymbol{v}}^r$ mit korrespondierender Kovarianz $\mathbf{P}_r$ auf Basis eines einzelnen Radarscans geschätzt werden. Somit ist kein Scan-Matching notwendig, was dieses Verfahren auch bei hoher Dynamik sehr robust macht, da eine sich stark ändernde Szene kein Problem darstellt. Robuste Bewegungsschätzung mit Scan-Matching funktioniert dagegen am besten bei geringen Änderungen der Szene zwischen nachfolgenden Scans, damit Objekte über möglichst viele Scans verfolgt werden können.

5.2.3 Lösung mit RANSAC-Least-Squares

Abbildung 5.2 zeigt auf der rechten Seite eine realistische Umgebung. Hier liegt ein Outlier, verursacht durch einen Messfehler (Objekt 1) sowie ein dynamisches Objekt (Objekt 3) vor. Diese können durch Mehrwegereflexionen oder auch Ghosting entstehen. Eine naive Lösung mit (5.3) würde hier eine suboptimale Lösung erzeugen, da alle Objekte genutzt werden würden. Daher wird das Verfahren weiterentwickelt, wie in Doer et al. [178] veröffentlicht.

Random Sample and Consenus (RANSAC) mit der Anwendung auf Least-Squares Parameterschätzung wurde von [40] vorgeschlagen. RANSAC-basierte Verfahren sind sehr robust, auch gegenüber einem hohen Anteil an Outliern. Es handelt sich hierbei um einen iterativen Ansatz. In jeder Iteration wird zufallsbasiert eine Stichprobe gezogen, die Modellparameter mittels der Stichprobe gelöst und alle Messungen bezüglich Inlier oder Outlier klassifiziert. Dies wird mehrmals wiederholt, wobei die Lösung mit dem größten Inlieranteil als finale Inliermenge bezeichnet wird. Die Anzahl an Iterationen n, die benötigt werden, um bei einer definierten Outlier-Wahrscheinlichkeit $p(\text{Outlier})$ mit einer Wahrscheinlichkeit $p(\text{Erfolg})$ die wahren Inlier zu identifizieren, lautet [40]:

$$n = \frac{\log\left(1 - p(\text{Erfolg})\right)}{\log\left(1 - p^K(\text{Inlier})\right)} = \frac{\log\left(1 - p(\text{Erfolg})\right)}{\log\left(1 - (1 - p(\text{Outlier}))^K\right)}, \tag{5.6}$$

wobei K die Anzahl an benötigten Messungen ist, um das Modell zu lösen.

Angewendet auf die Radar-Eigengeschwindigkeit ergibt sich der 3-Point RANSAC-LSQ Ansatz [178]. Es werden mindestens drei Messungen benötigt, um (5.2) zu lösen. Diese werden zufällig ausgewählt, um $\boldsymbol{v}^r_{\text{Hypthese}}$ zu bestimmen. Anschließend werden alle Messungen hinsichtlich Inlier oder Outlier evaluiert. Hierfür wird der Fehler $\boldsymbol{e}$ aus (5.5) für jede Messung aufgestellt. Ist der Fehlerbetrag $|\boldsymbol{e}|$ in allen Dimensionen unter einem Schwellwert, wird die Messung als Inlier klassifiziert. Das Verfahren ist sehr effizient. Bei einer Outlier-Wahrscheinlichkeit von 30 % können die Inlier mit einer Wahrscheinlichkeit von 99,9 % in nur 17 Iterationen nach (5.6) korrekt identifiziert werden. Im letzten Schritt werden (5.3) - (5.5) mit den als Inlier klassifizierten Messungen berechnet. Liegt die resultierende geschätzte Unsicherheit $\mathbf{P}_r$ oberhalb eines Schwellwerts, wird die Messung als Outlier klassifiziert und nicht weiter verarbeitet.

5.2.4 Lösung mit RANSAC-ODR

Als finalen Schritt der Eigengeschwindigkeitsschätzung kann eine nichtlineare Optimierung in Form einer Orthogonal Distance Regression (ODR) durchgeführt werden, wie von Kellner et al. im 2D-Fall vorgeschlagen wurde [67]. Die ODR berücksichtigt Unsicherheiten des Messmodells, was hier die abnehmende Güte der Winkelschätzung in Abhängigkeit des Azimut- und Elevationswinkels ist. Wie in (2.18) verdeutlicht, ist die Winkelschätzung nahe 0° am besten und verschlechtert sich asymptotisch gegen 90° [59]. Die Lösung von (5.3) modelliert den Azimut- und Elevationswinkel jedoch als fehlerfrei. ODR ermöglicht es mittels einer Gewichtung die Winkelunsicherheit auch im Messmodel zu berücksichtigen. Die Lösung aus der RANSAC-LSQ, siehe Abschnitt 5.2.3, wird dabei als initiale Lösung genutzt und mit ODR verbessert. Hierzu wird *ODRPACK* [13] als Toolbox verwendet. Neben der verbesserten Radar-Eigengeschwindigkeit $\hat{\boldsymbol{v}}^r$ umfasst das Ergebnis auch die Kovarianzmatrix $\mathbf{P}_r$. Dabei kann $\mathbf{P}_r$ in geschlossener Form aus der ODR-Lösung bestimmt werden [13].

5.2.5 Ergebnisse

Die Evaluation der Radar-Eigengeschwindigkeitsschätzung erfolgt im Folgenden zunächst in Simulation, um die drei Verfahren zu vergleichen. Anschließend erfolgt eine umfangreiche Demonstration mit Realläufen.

Simulation

Die Radar-Eigengeschwindigkeitsschätzung wird in diesem Abschnitt mittels einer realitätsnahen Simulation evaluiert. Dies wird in Form von Monte-Carlo Simulationen durchgeführt. Für jeden Durchlauf wird erst eine dreidimensionale Radar-Eigengeschwindigkeit innerhalb eines definierten Intervalls aus einer Gleichverteilung gezogen. Anschließend wird ein Radarscan erstellt. Die Größe des Radarscans, also die Anzahl an Scan-Punkten, wird aus einer Normalverteilung $\mathcal{N}(\mu, \sigma)$ gezogen, wobei der Mittelwert μ und die Varianz σ mittels der Realdatensätze bestimmt wurde. Für jedes Element des Radarscans wird der Azimut- und Elevationswinkel aus einer Gleichverteilung über

Kenngröße	Wert
Größe Radarscan	$\mu = 40$, $\sigma = 15$
Sichtbereich Azimut und Elevation	120°
Messrauschen Azimut und Elevation	$\sigma_1 = 1°$, $\sigma_2 = 10°$
Quantisierung Azimut und Elevation	2,8°
Messrauschen Dopplergeschwindigkeit	0,05 m/s
Quantisierung Dopplergeschwindigkeit	0,125 m/s
Outlier-Wahrscheinlichkeit	5 %

Tabelle 5.1: Wichtige Kenngrößen der Radar-Eigengeschwindigkeitssimulation

den gesamten Sichtbereich von 120° gezogen. Auf Basis einer ebenfalls von einer Gleichverteilung gezogenen Distanz, können die Kartesischen Koordinaten aller Scanobjekte berechnet werden. Unter Verwendung von (5.1) werden die Dopplergeschwindigkeiten aller Scan-Punkte aufgestellt. Dies resultiert im Referenzscan, der keine Messfehler aufweist. Um Realitätstreue zu erzielen, wird Messrauschen und die Radarsensorcharakteristik modelliert, um aus dem Referenzscan die Messung zu erzeugen. Hierzu wird additives Gaußsches Rauschen zur Messung der Distanz und der Dopplergeschwindigkeit hinzugefügt. Die Messung des Azimut- und Elevationswinkels wird ebenfalls mit additiven Gaußschen Rauschen beaufschlagt, das jedoch proportional zum gemessenen Winkel zunimmt. Das Messrauschen σ_m des Azimut- und Elevationswinkel wird daher mittels $\sigma_m = \sigma_1 + \sigma_2 \sin(\theta)$ in Abhängigkeit des gemessenen Winkels θ bestimmt. Damit wird die Charakteristik des realen Radarsensors hinsichtlich der Winkelmessungsgenauigkeit modelliert. Der verrauschte Azimut- und Elevationswinkel sowie die verrauschte Dopplergeschwindigkeit werden abschließend noch quantisiert. Hierbei wird die gleiche Schrittweite wie beim realen Sensor verwendet. Abschließend wird der Radarscan mit einer definierten Wahrscheinlichkeit mit Outliern versehen. Aus den verrauschten Messwerten für Azimut, Elevation, Distanz und Dopplergeschwindigkeit wird schließlich die Messung des Radarscan erstellt. Kenngrößen der eingesetzten Simulation sind in Tabelle 5.1 zusammengefasst.

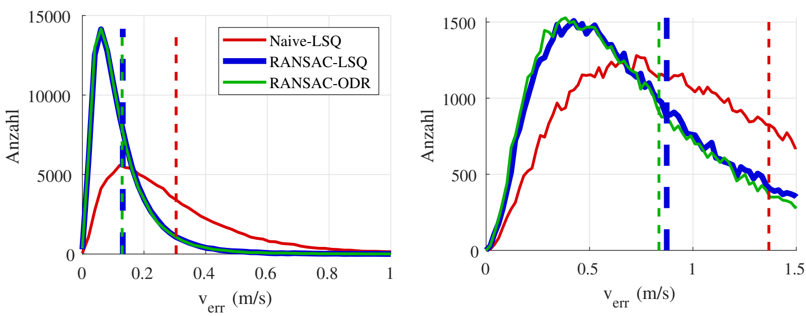
Die Simulation wurde für zwei Szenarien durchgeführt. Das Szenario *langsam* weist dabei eine maximale Geschwindigkeit von 2 m/s auf und deckt damit die Dynamik von Fußgängern sowie langsamen UAS-Flügen ab. Das Szenario *schnell*, weist eine maximale Geschwindigkeit von 20 m/s auf und evaluiert damit die Bewegung bei dynamischen Manövern oder dem Streckenflug größerer

UAS. Die Simulation wird mit Monte-Carlo Läufen durchgeführt, wobei für jeden Durchlauf die Referenz-Radar-Eigengeschwindigkeit aus einer Gleichverteilung mit definiertem Maximum (je nach Szenario: 2 m/s oder 20 m/s) gezogen wird. Zusätzlich wird der Radarscan auf Basis davon erstellt, verrauscht und anschließend von der Radar-Eigengeschwindigkeitsschätzung prozessiert. Dies wurde für 100 000 Durchläufe ausgeführt und für jeden Durchlauf der Geschwindigkeitsfehlerbetrag berechnet.

Es werden die drei Ansätze der Geschwindigkeitsschätzung verglichen:

- Naive-LSQ: Lösung mittels naiver LSQ unter Verwendung aller Objekte des Radarscans, wie in Abschnitt 5.2.2 beschrieben.
- RANSAC-LSQ: Lösung mittels RANSAC-LSQ zum Ausschließen von Outliern, wie in Abschnitt 5.2.3 erläutert.
- RANSAC-ODR: Lösung mittels RANSAC-ODR mit nachträglicher Optimierung des RANSAC-LSQ Ergebnisses, wie in Abschnitt 5.2.4 präsentiert.

Abbildung 5.3 zeigt das Histogramm des Geschwindigkeitsfehlers für das Szenario *langsam* (links) sowie für das Szenario *schnell* (rechts). Im Szenario *langsam* zeigt die Naive-LSQ mit 0,304 m/s einen deutlich höheren mittle-



(a) Simulationsszenario *langsam*

(b) Simulationsszenario *schnell*

Abbildung 5.3: Fehlerhistogramm des Geschwindigkeitsfehlerbetrags für das Szenario *langsam* (links) und *schnell* (rechts) für die drei Ansätze Naive-LSQ, RANSAC-LSQ, RANSAC-ODR. Die gestrichelten Linien markieren den Mittelwert des jeweiligen Geschwindigkeitsfehlerbetrags.

ren Fehler als die beiden RANSAC-Verfahren (RANSAC-LSQ: 0,131 m/s und RANSAC-ODR: 0,129 m/s). Zwischen der RANSAC-LSQ und der RANSAC-ODR zeigen sich beim Szenario *langsam* nur sehr geringe Unterschiede des Fehlers. Im Szenario *schnell*, erzielt die RANSAC-ODR mit 0,84 m/s das beste Ergebnis, dicht gefolgt von der RANSAC-LSQ mit 0,87 m/s. Wie auch schon beim Szenario *langsam* erzielt die Naive-LSQ mit einem mittleren Fehler von 1,4 m/s das schlechteste Ergebnis. Insgesamt zeigt sich beim Szenario *schnell* ein größerer Geschwindigkeitsfehler. Da im Vergleich zum Szenario *langsam* die maximale Geschwindigkeit um den Faktor zehn größer ist, ergeben sich auf Grund der Winkelschätzungsungenauigkeiten und Quantisierungseffekte größere Fehler der Geschwindigkeitsschätzung. Diese Unsicherheit der Winkelschätzung wird bei RANSAC-ODR modelliert, was zu einem besseren Ergebnis beim Szenario *schnell* führt. Die Verwendung des RANSAC-Verfahrens stellt einen effektiven Weg dar, um Outlier zu entfernen. So verbessert die RANSAC-LSQ den mittleren Geschwindigkeitsfehler um 57 % im Szenario *langsam* und um 34 % im Szenario *schnell*. Die Optimierung mittels ODR ergibt nur bei höheren Geschwindigkeiten (> 2 m/s) eine leichte Verbesserung. So erzielt die RANSAC-ODR im Szenario *schnell* einen um 3 % niedrigeren Fehler.

Evaluation mit Motion-Capture

Die Radar-Eigengeschwindigkeit wird in diesem Abschnitt mit Realläufen evaluiert. Dafür wurde die Multi-Sensorplattform aus Abschnitt 3.1 leicht modifiziert, indem der nach vorne gerichtete RISK-Radarsensor durch den AOP-Sensor ersetzt wurde. Der AOP-Sensor weist den gleichen Sichtbereich in Azimut und Elevation (120°) auf und erzielt daher bessere Ergebnisse insbesondere bezüglich der z^b-Komponente. Die Realläufe wurden in einem Motion-Capture-Lab aufgezeichnet und weisen daher eine sehr genaue Referenz auf. Die Referenzgeschwindigkeit wurde für die fünf Datensätze mit Hilfe der Referenzposen und der Inertialsensordaten mittels einer Batch-Optimierung mit der Toolbox *vicon2gt* [47] berechnet. Dies resultiert in einer Referenz für die Position, Lage und Geschwindigkeit des körperfesten Koordinatensystems. Dies ermöglicht die Evaluation der Eigengeschwindigkeit. Dafür muss jedoch die Radar-Eigengeschwindigkeit in das körperfeste Koordinatensystem transformiert werden, was unter Anwendung von (6.3) erreicht wird. Auf Grund des sehr kleinen Hebelarms von unter 0,1 m spielt das Messrauschen der Win-

kelgeschwindigkeit ω_m^b keine maßgebliche Rolle. So würde ein Rauschen der Winkelgeschwindigkeit von 0,5° zu einem Rauschen von $\boldsymbol{v}_w = 0{,}009\,\mathrm{m/s}$ führen. Da sich der Geschwindigkeitsfehler im Bereich von circa 0,13 m/s bewegt, kann dies daher vernachlässigt werden.

Abbildung 5.4 zeigt das Ergebnis für den Datensatz *Mocap-Difficult* und vergleicht die drei verschiedenen Schätzverfahren. Wie in Abbildung 5.4 zu erkennen ist, folgen alle drei Verfahren der Referenz sehr genau. Die Naive-LSQ weist jedoch Outlier zum Beispiel bei 18 s oder 103 s auf, die bei den beiden RANSAC-basierten Verfahren nicht auftreten. Zwischen der RANSAC-LSQ und der RANSAC-ODR Lösung lassen sich im Plot fast keine Unterschiede feststellen. Alle drei Geschwindigkeitskomponenten weisen ein ähnliches Verhalten bezüglich der Schätzung auf.

Zur weiteren Evaluation zeigt Abbildung 5.5 den Fehler zwischen der Referenz und der geschätzten Geschwindigkeit für alle drei Verfahren. Zur besseren Übersichtlichkeit ist die y^b-Achse auf den Fehlerbereich ±0,6 m/s begrenzt. Insbesondere die y^b- und z^b-Achse zeigt einen deutlich höheren Anteil an Outliern der Naiven-LSQ. Durch den RANSAC-Ansatz werden diese jedoch effektiv entfernt. Die anschließende Optimierung durch die ODR führt jedoch zu keiner erkennbaren Verbesserung. Daher weisen die blaue (RANSAC-LSQ) und die grüne (RANSAC-ODR) den annähernd gleichen Fehlerverlauf auf. Das sehr ähnliche Ergebnis zeigt sich auch bei Analyse des mittleren Geschwindigkeitsfehlerbetrags. So erzielt die Naive-LSQ: 0,35 m/s, die RANSAC-LSQ 0,15 m/s und die RANSAC-ODR 0,15 m/s.

Für eine umfassende Evaluation wurden insgesamt fünf Datensätze prozessiert. Diese weisen unterschiedliche Dynamiken auf, die maximale Geschwindigkeit ist jedoch auf Grund der limitierenden Größe des Motion-Capture-Labs auf circa 2,5 m/s begrenzt. Eine detaillierte Beschreibung erfolgt im Rahmen des Kapitels 6 im Abschnitt 6.1.5. Der mittlere Geschwindigkeitsfehlerbetrag für alle Datensätze und die drei Schätzverfahren sind in Tabelle 5.2 zusammengefasst. Es zeigt sich im Schnitt eine Verbesserung des mittleren Geschwindigkeitsfehler bei Verwendung von RANSAC um 38 %. Die nachfolgende Optimierung mit ODR erzielt in diesen Szenarien den gleichen mittleren Geschwindigkeitsfehler. Dies spiegelt die Ergebnisse der Simulation wider, da es sich auch hier um niedrige Geschwindigkeiten von unter 2,5 m/s handelt und somit der Fehler der Winkelschätzung des Radarscans zu vernachlässigbaren Geschwindigkeitsfehlern führt.

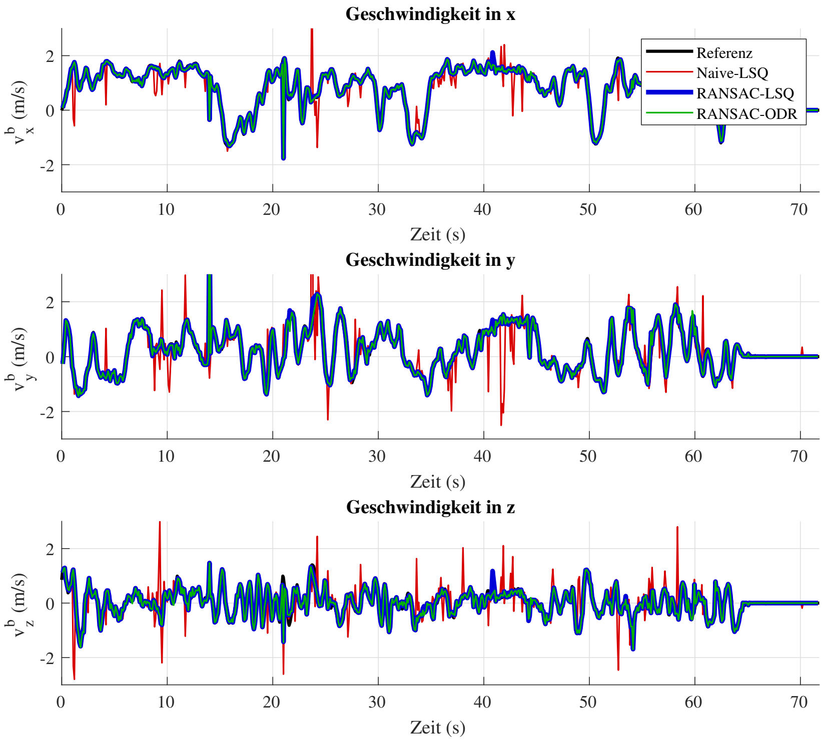


Abbildung 5.4: Verläufe der Geschwindigkeitskomponenten des Datensatzes *Mocap-Difficult* für die drei Schätzverfahren Naive-LSQ (rot), RANSAC-LSQ (blau) und RANSAC-ODR (grün) im Vergleich zur Referenz (schwarz).

Die Analyse der mittleren Laufzeit zeigt die hohe Effizienz dieses Ansatzes. So benötigt die RANSAC-LSQ für die Verarbeitung eines Radarscans durchschnittlich 0,3 ms auf dem *Intel-NUC-i7* und 0,53 ms auf dem *Up-Core*. Im Vergleich zur Naiven-LSQ, benötigt die RANSAC-LSQ zwar etwa die Vierfache Laufzeit, was aber durch die Zunahme der Genauigkeit gerechtfertigt ist. Auf Grund der nachfolgenden Optimierung bei der RANSAC-ODR, entsteht hier die größte mittlere Laufzeit.

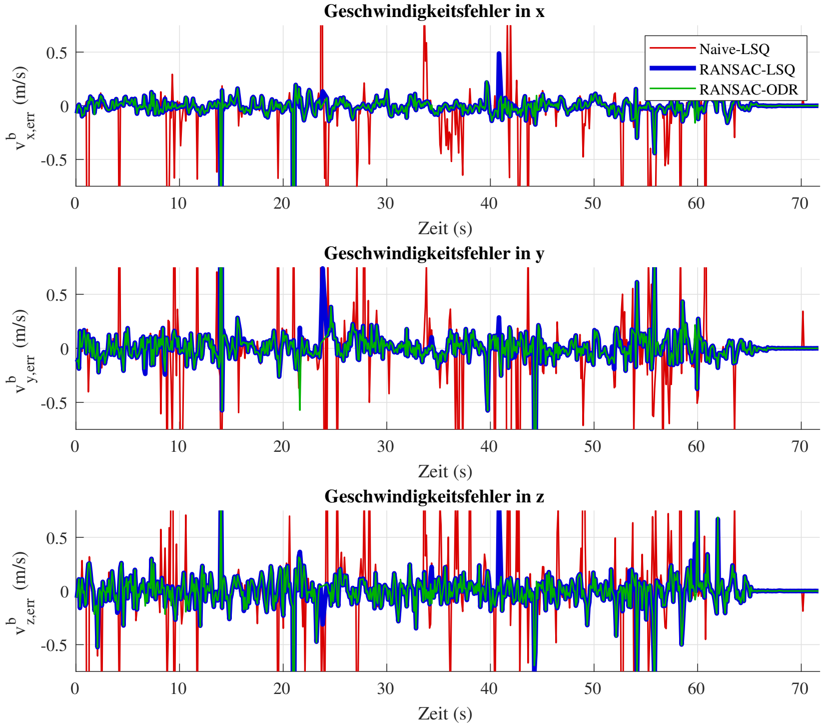


Abbildung 5.5: Verläufe der Geschwindigkeitsfehlerkomponenten des Datensatze *Mocap Difficult* für die drei Schätzverfahren Naive-LSQ (rot), RANSAC-LSQ (blau) und RANSAC-ODR (grün).

Datensatz	Naive-LSQ	RANSAC-LSQ	RANSAC-ODR
MoCap Easy	0, 10	0, 07	0, 07
MoCap Medium	0, 16	0, 11	0, 11
MoCap Difficult	0, 35	0, 15	0, 15
MoCap Dark	0, 20	0, 15	0, 16
MoCap Dark Fast	0, 26	0, 15	0, 16
Mittelwert	0, 21	0, 13	0, 13
Mittlere Laufzeit (ms)	0,03 (0,12)	0,13 (0,53)	0,28 (1,48)

Tabelle 5.2: Evaluation des mittleren Geschwindigkeitsfehlerbetrags (m/s) der fünf Motion-Capture Datensätze für die drei Schätzverfahren. Die mittlere Laufzeit für die Verarbeitung eines Radarscans ist für das *Intel-NUC-i7* und das *Up-Core* in Klammern angegeben.

Metrik	RANSAC-LSQ	RANSAC-ODR
Mittelwert $\boldsymbol{\mu}$ (m/s)	[0,005 −0,003 −0,002]	[0,005 −0,003 −0,003]
Standartabweichung $\boldsymbol{\sigma}_r$ (m/s)	[0,06 0,09 0,10]	[0,06 0,09 0,10]

Tabelle 5.3: Spezifizierung der Schätzfehler hinsichtlich des Mittelwerts und der Standardabweichung für Realdaten.

Abschließend erfolgt die Evaluation hinsichtlich der stochastischen Beschreibung des Geschwindigkeitsfehlers. Dabei werden sowohl Mittelwert als auch die Standardabweichung der Geschwindigkeitsfehler aller drei Dimensionen bestimmt. Im Schnitt über alle fünf Datensätze ergeben sich die Ergebnisse in Tabelle 5.3. Auch hier zeigt sich kein deutlicher Unterschied zwischen RANSAC-LSQ und RANSAC-ODR sowohl beim Mittelwert als auch der Standardabweichung. Beide Verfahren erzielen einen Mittelwert von maximal 0,005 m/s (in der x-Komponente). Daher wird die Eigengeschwindigkeitsschätzung als annähernd mittelwertfrei eingestuft. Die Analyse der Standardabweichung zeigt im Schnitt 0,06 m/s (x-Komponente), 0,09 m/s (y-Komponente) und 0,10 m/s (z-Komponente). Dies lässt sich durch den Sichtbereich sowie den typischen Szenen der Datensätze erklären. Da der Roll- und Pitch-Winkel der Plattform meist nahe 0° lag, ist das körperfeste-Koordinatensystem und, auf Grund des Hardwareaufbaus auch das Radar-Koordinatensystem, meist zur Erdbeschleunigung ausgerichtet. Somit sind auf Grund der Szene, dem cica 10×10 m großen Motion-Capture-Lab, der Radarscan horizontal viel stärker aufgespannt als vertikal. Dies führt zu einer schlechteren Schätzung der z-Komponente im Vergleich zur x-Komponente. Der AOP-Sensor weist einen Sichtbereich von 120° in Azimut auf. Dadurch kann die x-Komponente besser als die y-Komponente geschätzt werden.

Die Werte für Mittelwert und Standardabweichung werden im weiteren Verlauf der Arbeit zur realistischen Simulation des Radar-inertialen Navigationsfilters mit Hilfe einer Normalverteilung $\mathcal{N}(\mathbf{0}, \boldsymbol{\sigma}_r)$ verwendet. Da im Bereich niedriger Geschwindigkeiten kein klarerer Unterschied zwischen der RANSAC-LSQ und RANSAC-ODR erkennbar ist, wird zum Sparen von Rechenzeit das effizientere Verfahren RANSAC-LSQ bei Geschwindigkeiten unterhalb von 4 m/s im weiteren Verlauf der Arbeit eingesetzt.

5.3 Radar-Gierwinkelschätzung

Neben der Eigengeschwindigkeitsmessung kann auch eine Gierwinkel-Schätzung mit Hilfe eines einzelnen Radarscans auf Basis von Manhattan-Welt Annahmen durchgeführt werden. Dies wurde zunächst unter der Verwendung von einem einzelnen Radar in Doer et al. [181] veröffentlicht und für Multi-Radar-Setups mit Doer et al. [182] erweitert. Manhattan-Welt Annahmen wurden bereits erfolgreich im Bereich Computer Vision beispielsweise für einen visuellen Kompass [29] oder zur Verbesserung der Rekonstruktion in Innenräumen [41] eingesetzt.

Radar hat das fundamentale Messprinzip, dass der orthogonale Abstand zu flächigen Objekten oder Ebenen mit glatter Oberfläche gemessen wird. Daher kann, über den Azimut- und Elevationswinkel des Radarscan-Punktes, die Lage der Ebene relativ zum Radarsensor bestimmt werden. Abbildung 5.6 zeigt dies anschaulich mit einem zweidimensionalen Beispiel. Mittels das Azimut-Winkels $\theta_{\text{A,Ebene}}$ kann der Gierwinkel der Ebene γ_{Ebene} aufgestellt werden. Unter Einbeziehung des Navigationsfilterzustands, siehe Unterabschnitt 5.4.1, kann die Lage sogar global bestimmt werden. Um diese Eigenschaft effektiv zur Gierwinkel-Stützung einzusetzen, sind Annahmen an die Umgebung notwendig.

5.3.1 Annahmen an die Umgebung

Die Umgebung wird als weitgehend konsistent mit einer zweidimensionalen Manhattan-Welt angenommen. Das heißt, ebene Strukturen sind rechtwinklig zueinander, wie in Abbildung 5.7 dargestellt. Dabei zeigt der linke Teil einen Ausschnitt eines Grundrisses und der rechte Teil einen Ausschnitt des KIT

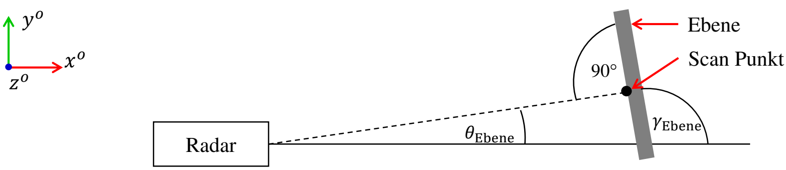
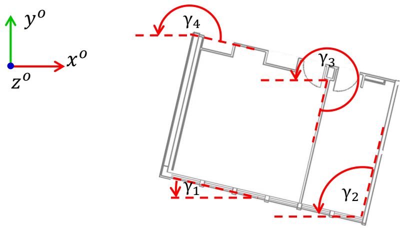


Abbildung 5.6: Radar-Charakteristik an Ebenen



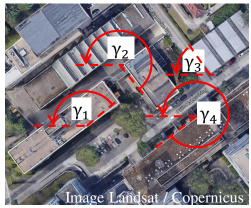


Abbildung 5.7: Beispiel für Innenraum (links) und urbane Gebiete (rechts), die durch eine Manhattan-Welt beschreibbar sind.

Campus. Alle Manhattan-Welt Winkel γ_i mit $i \in 1, 2, 3, 4$ sind orthogonal zueinander. Typische Manhattan-Welt Strukturen sind dabei vor allem Wände sowohl in Innenräumen als auch in urbanen Teilen. Insbesondere in Innenräumen sind große, flächige Objekte wie beispielsweise Möbel auch in der Regel parallel zu Wänden aufgestellt und folgen daher der Manhattan-Welt Annahme.

Des Weiteren wird angenommen, dass alle Manhattan-Welt-Objekte zur Erdbeschleunigung ausgerichtet sind („im Wasser stehen“). Dies ist für bebaute Strukturen in der Regel der Fall und daher ein wichtiges Kriterium. Neben Wänden wird dies meist auch von Objekten wie Möbel, Fenster und Türen erfüllt. Alle Objekte, die diese Annahmen erfüllen, werden im Folgenden Manhattan-Welt-Objekte (MWO) genannt.

Wie in Abbildung 5.7 dargestellt, ist die Messung eines MWO nicht eindeutig, da eine Zuordnung zu einem der vier globalen Manhattan-Welt Winkel γ_i nicht möglich ist. Daher wird in dieser Arbeit die Ausrichtung der Manhattan-Welt γ_m mittels Modulo Operation

$$\gamma_m = \mathrm{mod}(\gamma_i, 90^\circ) \tag{5.7}$$

durch das Intervall $[0^\circ, 90^\circ]$ begrenzt. Die Gierwinkelstützung nutzt den Inlier-Radarscan aus der Geschwindigkeitsschätzung, da dieser die dynamischen Objekte sowie fehlerhafte Messungen durch den RANSAC-Ansatz nicht mehr aufweist. Es sind zwei Schritte notwendig, wie in Abbildung 5.8 illustriert ist. Zunächst muss eine Initialisierung durchgeführt werden, um die Ausrichtung γ_0^n der Manhattan-Welt bezüglich des Navigationskoordinatensystem zu bestimmen. Anschließend können Gierwinkel-Messungen erzeugt werden, wobei der aktuelle Filterzustand genutzt wird, um eine eindeutige Gierwinkelmessung zu

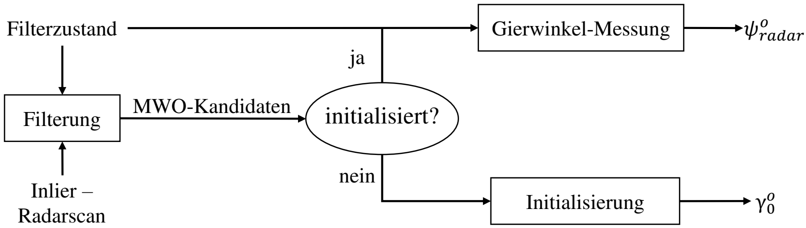


Abbildung 5.8: Blockdiagramm Gierwinkelstützung mittels Manhattan-Welt Annahmen

erzeugen. Dies baut auf der Annahme auf, dass der Gierwinkelfehler kleiner als 45° ist, um die korrekte Zuordnung zum korrespondierenden Manhattan-Welt-Winkel γ_i durchführen zu können. Da die Gierwinkeldrift typischerweise im einstelligen Gradbereich liegt, wird dies als eine valide Annahme eingestuft. Im Folgenden werden die Blöcke Filterung, Initialisierung und Gierwinkel-Messung aus Abbildung 5.8 näher erläutert.

5.3.2 Vorfilterung des Radarscans

Die Filterung des Radarscans erfolgt in mehreren Schritten. Zunächst wird eine Bewegungsrandbedingung gefordert, da beispielsweise bei der Initialisierung während des Stillstandes keine neuen Informationen zur Umgebung gesammelt werden. Dann werden Objekte herausgefiltert, die ein zu schlechtes SNR haben oder zu weit weg sind. Anschließend werden alle übrigen Radarscan-Punkte bezüglich der Ausrichtung zur Erdbeschleunigung evaluiert. Hierfür wird zunächst der Radarscan Punkt $\boldsymbol{p}^r$ ins stabilisierte Koordinatensystem, siehe Unterabschnitt 2.1, transformiert. Mit Hilfe von Roll θ_{ob}, Pitch ϕ_{ob} und der Rotation vom Radar zum körperfesten Koordinatensystem $\mathbf{C}^b_r$ aus dem Systemzustand (5.17) folgt für den Radarscan Punkt im stabilisierten Koordinatensystem $\boldsymbol{p}^s$:

$$\boldsymbol{p}^s = \mathbf{C}(\theta_{ob}, \phi_{ob})\ \mathbf{C}^b_r\ \boldsymbol{p}^r\,. \tag{5.8}$$

Transformation in Kugelkoordinaten ergibt:

$$r_s = \sqrt{p_x^s p_x^s + p_y^s p_y^s + p_z^s p_z^s}, \tag{5.9}$$

$$\theta_A = \operatorname{atan2}(p_y^s, p_x^s), \tag{5.10}$$

$$\phi_A = \operatorname{atan2}(\sqrt{p_x^s p_x^s + p_y^s p_y^s}, p_z^s). \tag{5.11}$$

Der Radarscan Punkt wird folglich als Manhattan-Welt-Objekt-Kandidat (MWO-Kandidat) eingestuft, falls

$$|\theta_A| - \pi/2 < \theta_{\text{thresh}} \tag{5.12}$$

gilt. Alle MWO-Kandidaten werden entweder zur Initialisierung oder zur Generierung der Gierwinkelmessung eingesetzt.

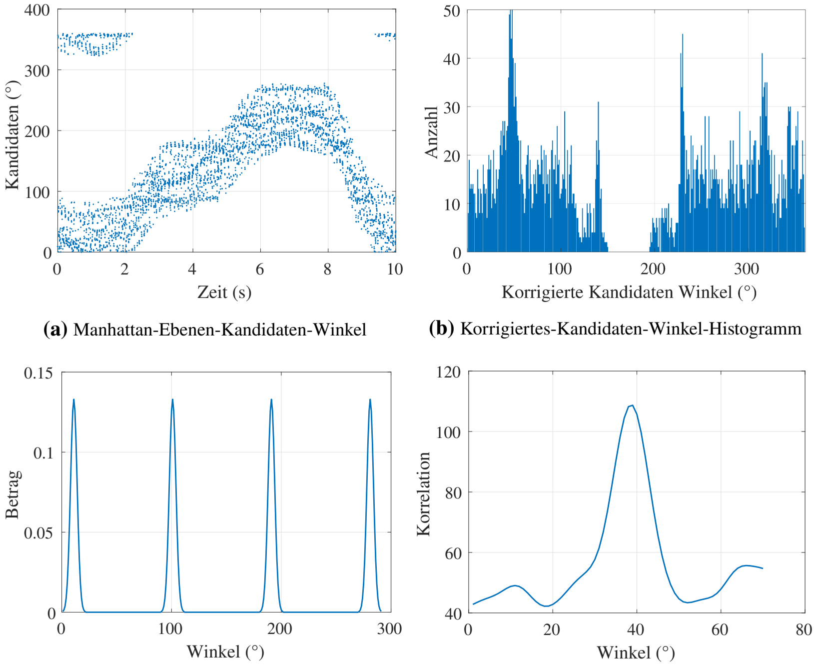
5.3.3 Initialisierung

Da die Ausrichtung γ_0^n der Manhattan-Welt bezüglich des Navigationskoordinatensystems zunächst unbekannt ist, muss diese initialisiert werden. Während der Initialisierung wird der Gierwinkel nur durch Integration der Winkelgeschwindigkeiten im Strapdown-Algorithmus berechnet. Da die Initialisierung in der Regel nur 10 – 20 s dauert, kann die Gierwinkeldrift während dieser Zeit vernachlässigt werden. Alle Kandidaten werden zunächst mit

$$\boldsymbol{p}^o = \mathbf{C}_b^o \, \mathbf{C}_r^b \, \boldsymbol{p}^r \tag{5.13}$$

ins Navigationskoordinatensystem transformiert werden. Mit Hilfe von (5.10) wird der Kandidaten-Winkel $\theta_{A,s}$ berechnet. Abbildung 5.9a zeigt die Kandidaten-Winkel über einen Zeitraum von 10 s wobei alle drei Radarsensoren, wie in Abbildung 3.3 gezeigt, verwendet wurden.

Anschließend wird das Histogramm der Kandidaten-Winkel berechnet. Für ein robusteres Ergebnis wird dieses anschließend so verschoben, dass das Maximum bei 45° liegt. Hierbei wird mittels des Modulo-Operators der Wertebereich entsprechend begrenzt. Das verschobene Histogramm ist in Abbildung 5.9b gezeigt.



(a) Manhattan-Ebenen-Kandidaten-Winkel

(b) Korrigiertes-Kandidaten-Winkel-Histogramm

(c) Korrelationsmaske

(d) Korrelation Histogramm und Maske

Abbildung 5.9: Schritte der Manhattan-Winkel-Initialisierung nach Doer et al. [182]

Die Kandidaten werden als konsistent mit einer Manhattan-Welt angenommen. Somit folgt für das Histogramm die Annahme, dass sich Maxima im Abstand von 90° zeigen und somit γ_i mit $i \in 1, 2, 3, 4$ aus Abbildung 5.7 abbilden. Da hier jedoch realistische Umgebungen betrachtet werden, sind diese Maxima nicht immer klar ausgebildet, wie im Winkelhistogramm in Abbildung 5.9b zu erkennen ist. Um Fehldetektionen herauszufiltern, wird ein Korrelationsansatz verfolgt. Dabei wird das Winkelhistogramm mit der Korrelationsmaske aus Abbildung 5.9c korreliert. Dies ergibt das Plot in Abbildung 5.9d mit einem klar ausgeprägtem Maximum. Im Gegensatz zu einer Bestimmung von γ_{approx} nur auf Basis des Winkelhistogramms ist die Bestimmung mittels Korrelation deutlich robuster. Ist das Maximum oberhalb eines zuvor definierten Schwellwerts, kann initialisiert werden.

Als finaler Schritt werden dann alle Kandidaten bezüglich Inlier oder Outlier klassifiziert. Hierfür wird die Differenz d zu γ_{approx} mit

$$d = |\ \text{mod}(\theta_{A,s} + 90^\circ - \gamma_{\text{approx}}, 90^\circ) - 45^\circ\ | \tag{5.14}$$

ausgewertet. Die Ausrichtung der Manhattan-Welt im Navigationskoordinatensystem γ_0^n ergibt sich schließlich aus dem Mittelwert aller Inlier.

5.3.4 Generierung der Gierwinkel Messung

Sobald die Initialisierung abgeschlossen ist, können Gierwinkelmessungen ψ^n_{radar} generiert werden. Dafür werden alle MWO-Kandidaten mit (5.13) ins Navigationskoordinatensystem transformiert. Anschließend wird der globale Azimut-Winkel mittels (5.10) und der aktuellen Navigationslösung berechnet. Somit kann die Winkeldifferenz der Manhattan-Welt $d_{\text{Manhattan}}$

$$d_{\text{Manhattan}} = |\ \text{mod}\left(\theta_{A,s} - \gamma_0 + 45^\circ,\ 90^\circ\right) - 45^\circ\ | \tag{5.15}$$

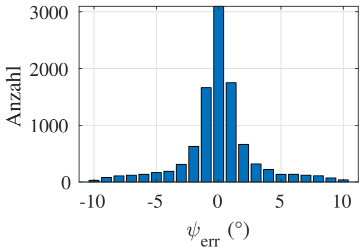
aufgestellt werden. Aus allen MWO-Kandidaten wird der Kandidat mit der geringsten Differenz $d_{\text{Manhattan}}$ ausgewählt und zur Gierwinkelberechnung genutzt. Zunächst erfolgt die Transformation in das stabilisiertes Koordinatensystem mit (5.8). Dann wird mit (5.10) in Kugelkoordinaten transformiert, was den Azimut-Winkel $\theta_{A,s}$ ergibt. Die globale Gierwinkelmessung $\psi_{ob,\text{radar}}$ wird wie folgt aufgestellt:

$$\psi_{ob,\text{radar}} = \theta_{A,s} + \gamma_0 + k\ \pi/2, \tag{5.16}$$

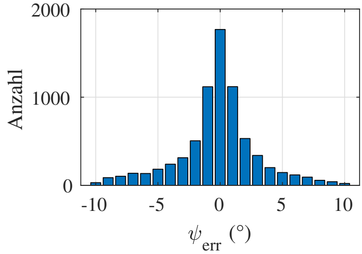
wobei $k \in \{0, 1, 2, 3\}$, da nicht bekannt ist welcher der vier Manhattan-Welt Winkel γ_i mit $i \in 1, 2, 3, 4$ aus Abbildung 5.7 observiert wird. Mit Hilfe des erwarteten Gierwinkels ψ_{ob} aus dem Filterzustand (5.17) kann k berechnet werden.

5.3.5 Ergebnisse

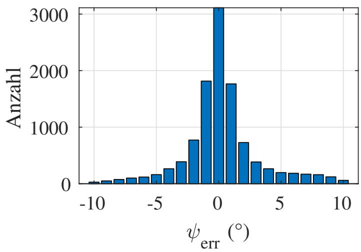
Die Evaluation der Radar-Gierwinkelstützung erfolgt mit vier Indoor-Datensätzen, die zur ausführlichen Evaluation der Radar-inertialen Navigation im weiteren Verlauf des Kapitels, in Abschnitt 5.4.5, genutzt werden. Für eine ausführliche Beschreibung der Datensätze sowie zu Erstellung der Referenz sei daher auf Abschnitt 5.4.5 verwiesen. Da für alle vier Datensätze eine Referenz vorliegt, kann der Gierwinkel evaluiert werden. Hierfür wurde erst die Initialisierung gemäß Abschnitt 5.3.3 in den ersten 10 s der Datensätze durchgeführt. Anschließend wurden die Gierwinkelmessungen, wie in Abschnitt 5.3.4 beschrieben, erzeugt. Auf Basis der Referenz kann dann der Messfehler bestimmt werden. Die Histogramme der Messfehler sind in Abbildung 5.10 für alle vier Datensätze dargestellt. Dabei wurden die Messfehler nur innerhalb



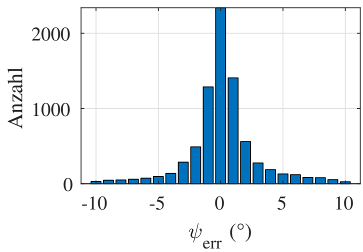
(a) *Office-Floor*



(b) *Lab-Floor*



(c) *Workshop*



(d) *Basement*

Abbildung 5.10: Analyse des geschätzten Gierwinkelfehlers für vier Indoor-Datensätze. Mit Hilfe der Referenz konnte der jeweilige Fehler bestimmt werden und wird mittels eines Histogramms veranschaulicht. Die Fehler sind auf den Bereich $\pm 10°$ begrenzt.

des Intervalls $\pm 10°$ betrachtet. Fehler, die größere Werte annehmen, würden von dem Outlier-Test des Kalman-Filters herausgefiltert werden. Alle vier Histogramme weisen einen ähnlichen Verlauf auf. Dies zeigt, dass die Radar-Gierwinkelstützung auch in unterschiedlichen realistischen Umgebungen mit Objekten, Möbeln sowie Werkzeugmaschinen robust eingesetzt werden kann und eine Winkelmessung von hoher Güte liefert. Der Mittelwert des Gierwinkelfehlers über alle vier Datensätze beträgt 0,05°, kann also approximativ als mittelwertfrei eingestuft werden. Die mittlere Standardabweichung beträgt 2,9° und stellt somit eine sehr genaue Gierwinkelmessung dar.

5.4 Navigationssystem

Zuvor wurde die Radar-basierte Schätzung der Eigengeschwindigkeit und Gierwinkel erläutert. Diese Informationen werden in einem Navigationssystem mit den Sensordaten einer IMU in einem Error-State-Kalman-Filter, siehe Unterabschnitt 2.2.2, fusioniert. Daraus resultiert ein effizientes und genaues Navigationssystem, was eine Navigationslösung auch in erschwerten Sichtbedingungen liefert. Das folgende Navigationssystem ist iterativ mit den Publikationen Doer et al. [178, 179, 181, 182] entwickelt worden. Zunächst wurde nur die Fusion von Inertialsensorik und der Radar-Eigengeschwindigkeit mit nur einem Radarsensor untersucht [178]. Dieses System wurde dann erweitert, um auch die extrinsische Kalibrierung, also die Translation und Rotation des Radarkoordinatensystems bezüglich des körperfesten Koordinatensystems, online zu schätzen [179]. Da der akkumulierende Positionsfehler primär auf eine Gierwinkeldrift zurückzuführen war, erfolgte die Erweiterung mit der Gierwinkelstützung [181]. Schließlich verallgemeinert [182] den Ansatz auf eine frei parametrierbare Anzahl an Radarsensoren mit online Kalibrierung, Gierwinkelstützung und Laufzeitoptimierung. Die nachfolgende Systembeschreibung folgt dem zuletzt genannten verallgemeinerten Ansatz aus [182].

5.4.1 Filterarchitektur

Filterzustand

Der Basis-Systemzustand zum Zeitschritt k ist durch

$$\boldsymbol{x}_{\mathrm{k,Basis}} = \begin{bmatrix} \boldsymbol{x}_{k,\mathrm{IMU}}^T & \boldsymbol{x}_{k,\mathrm{Radare}}^T \end{bmatrix}^T \tag{5.17}$$

definiert und besteht auf dem IMU-Zustand $\boldsymbol{x}_{k,\mathrm{IMU}}$ und dem Radar-Kalibrierungszustand $\boldsymbol{x}_{k,\mathrm{Radare}}$. Der IMU-Zustand

$$\boldsymbol{x}_{k,\mathrm{IMU}} = \begin{bmatrix} \boldsymbol{p}_{ob}^{o,T} & \boldsymbol{v}_{ob}^{o,T} & \boldsymbol{q}_{b}^{o,T} & \boldsymbol{b}_{a}^{b,T} & \boldsymbol{b}_{g}^{b,T} & b_b \end{bmatrix}^T \tag{5.18}$$

besteht aus der Position $\boldsymbol{p}_{ob}^{o}$, Geschwindigkeit $\boldsymbol{v}_{ob}^{o}$ und Lage $\boldsymbol{q}_{b}^{o}$ im lokalen Koordinatensystem, sowie dem Offset des Beschleunigungssensors $\boldsymbol{b}_{a}^{b}$ und des Drehratensensors $\boldsymbol{b}_{g}^{b}$ im körperfesten Koordinatensystem und dem Barometeroffset b_b. Ist ein Setup aus L-Radarsensoren gegeben, folgt für den Radar-Kalibrierungszustand $\boldsymbol{x}_{k,\mathrm{Radare}}$:

$$\boldsymbol{x}_{k,\mathrm{Radare}} = \begin{bmatrix} \boldsymbol{x}_{k,\mathrm{Radar,1}}^T & \dots & \boldsymbol{x}_{k,\mathrm{Radar,L}}^T \end{bmatrix}^T, \tag{5.19}$$

$$\boldsymbol{x}_{k,\mathrm{Radar,l}} = \begin{bmatrix} \boldsymbol{l}_{br,l}^{b,T} & \boldsymbol{q}_{r,l}^{b,T} \end{bmatrix}^T, \tag{5.20}$$

wobei $\boldsymbol{l}_{br,l}^{b}$ der translatorische und $\boldsymbol{q}_{r,l}^{b}$ der rotatorische Teil der Transformation vom Radarkoordinatensystem $\{\}^r$ des l-ten Radarsensors in das körperfeste Koordinatensystem ist.

Initialisierung

Bei Initialisierung wird ein Stillstand der Multisensorplattform über den gesamten Initialisierungszeitraums (typischerweise 10 s) angenommen. Während dieser Zeit werden die Messdaten der IMU gesammelt. Der Drehratenoffset $\boldsymbol{b}_g$ wird dann mit dem Mittelwert aller Drehratenmessungen ω_{ib}^{b} initialisiert. Gleiches gilt für den Barometeroffset b_b, dieser wird mit der mittleren Höhe aus (5.46) initialisiert. Da die initiale Ausrichtung im Allgemeinen nicht

bekannt ist, wird diese mit $\psi_{ob,0} = 0°$ initialisiert. Unter der Annahme, dass die Offsets des Beschleunigungsmessers im Vergleich zur Erdbeschleunigung klein sind, kann der initiale Rollwinkel θ_{ob} und Pitchwinkel ϕ_{ob} unter Nutzung der mittleren gemessenen Beschleunigung $\overline{\boldsymbol{a}}_m^b$ mit

$$\theta_{ob} = -\text{atan2}\left(\overline{a}_{m,y}^b, -\overline{a}_{m,z}^b\right), \tag{5.21}$$

$$\phi_{ob} = \arcsin\left(\frac{\overline{a}_{m,y}^b}{g}\right) \tag{5.22}$$

initialisiert werden. Hierbei bezeichnet atan2 die Arkustangens-Funktion mit automatischer Korrektur des Quadranten.

Zustandserweiterung

Auf Grund von Verzögerungen bei der Verarbeitung der Radar-Rohdaten auf dem Radarchip und durch die Übertragung entstehen bei den eingesetzten Sensoren Verzögerungen bis zu 90 ms. Diese Verzögerungen können insbesondere bei hoher Dynamik nicht vernachlässigt werden. Ein naiver Ansatz wäre das Navigationssystem leicht verzögert, zum Beispiel mit einer Latenz von 100 ms, zu prozessieren. Das Ziel dieser Arbeit ist jedoch ein Echtzeitnavigationssystem. Daher erfolgt die Verarbeitung der verzögerten Radar-Messungen durch Zustandserweiterung mittels Stochastic-Cloning, siehe Unterabschnitt 2.2.2.

Da die Multisensorplattform, siehe Unterabschnitt 3.1, die Radarsensoren aktiv triggert, ist der exakte Zeitpunkt einer Messung relativ zu den IMU-Messungen bekannt. Daher wird der Zustand mit dem Empfang des Trigger-Signals des l-ten Radarsensors wie in [182]

$$\boldsymbol{x}_{k,\text{erweitert}} = \begin{bmatrix} \boldsymbol{x}_k^T & \boldsymbol{x}_{\{\text{voll},\text{approx}\},l}^T \end{bmatrix}^T \tag{5.23}$$

erweitert. Dabei sind zwei verschiedene Methoden der Zustandserweiterung vorgeschlagen worden [182]. Der vollständige Klon $\boldsymbol{x}_{\text{voll},l}$ beinhaltet alle Zu-

stände, die für die Fusion der Radar-Eigengeschwindigkeit und Gierwinkelmessung benötigt werden und ist definiert durch:

$$\boldsymbol{x}_{\text{voll},l} = \begin{bmatrix} \boldsymbol{p}_{ob}^{o,T} & \boldsymbol{v}_{ob}^{o,T} & \boldsymbol{q}_{b}^{o,T} & \boldsymbol{b}_{g}^{b,T} & \boldsymbol{l}_{br,l}^{b,T} & \boldsymbol{q}_{r,l}^{b,T} \end{bmatrix}^{T} . \tag{5.24}$$

Der approximierte Klon $\boldsymbol{x}_{\text{approx},l}$ nimmt dagegen an, dass sich der Zustand des Drehratenoffsets $\boldsymbol{b}_g^b$ und der Radar-Kalibrierung bestehend aus $\boldsymbol{l}_{br,l}^b$ und $\boldsymbol{q}_{r,l}^b$ zwischen der Zustandserweiterung und dem Filterupdate nicht ändern. Der approximierte Klon ist somit kleiner, was Rechenzeit spart und ist wie folgt definiert:

$$\boldsymbol{x}_{\text{approx},l} = \begin{bmatrix} \boldsymbol{p}_{ob}^{o,T} & \boldsymbol{v}_{ob}^{o,T} & \boldsymbol{q}_{b}^{o,T} \end{bmatrix}^{T} . \tag{5.25}$$

Neben dem Systemzustand wird auch die Kovarianzmatrix durch

$$\mathbf{P}_{k,\text{erweitert}} = \begin{bmatrix} \mathbf{I}_N \\ \mathbf{J}_{\{\text{voll},\text{approx}\}} \end{bmatrix} \mathbf{P}_k \begin{bmatrix} \mathbf{I}_N & \mathbf{J}_{\{\text{voll},\text{approx}\}}^T \end{bmatrix} \tag{5.26}$$

erweitert. Abhängig vom Typ der Zustandserweiterung ergibt sich bei Zustandsgröße N:

$$\tag{5.27}$$

$$\mathbf{J}_{\text{voll}} = \begin{bmatrix} \mathbf{I}_9 & \mathbf{0}_{9\times3} & 0 & \mathbf{0}_{9\times3} & \mathbf{0}_{9\times n_1} & \mathbf{0}_{9\times6} & \mathbf{0}_{9\times n_2} \\ \mathbf{0}_{3\times6} & \mathbf{0}_{3\times3} & 0 & \mathbf{I}_3 & \mathbf{0}_{3\times n_1} & \mathbf{0}_{3\times6} & \mathbf{0}_{3\times n_2} \\ \mathbf{0}_{6\times6} & \mathbf{0}_{6\times3} & 0 & \mathbf{0}_{6\times3} & \mathbf{0}_{6\times n_1} & \mathbf{I}_6 & \mathbf{0}_{6\times n_2} \end{bmatrix}, \tag{5.28}$$

$$\mathbf{J}_{\text{approx}} = \begin{bmatrix} \mathbf{I}_9 & \mathbf{0}_{9\times(N-9)} \end{bmatrix} . \tag{5.29}$$

Dabei gilt $n_1 = 18(l-1)$ und $n_2 = 18(L-l)$ unter der Annahme von L-Radarsensoren. Sobald die korrespondierenden Filterupdates abgeschlossen sind, wird der Klon vom Systemzustand entfernt, um Rechenzeit zu sparen.

5.4.2 Radar-Geschwindigkeitsupdate

Das Radar-Geschwindigkeitsupdate fusioniert die geschätzte Radar-Eigengeschwindigkeit $\boldsymbol{v}^r_{m,l}$ mit Kovarianz $\mathbf{P}^r_l$ des l-ten Radarsensors aus Unterabschnitt 5.2. Dafür muss zunächst die Messgleichung aufgestellt werden. Dies erfolgt hier für den allgemeinen Fall von N-Radarsensoren [182].

Unter der Annahme, dass der Radarsensor starr mit der IMU, also dem körperfesten Koordinatensystem, verbunden ist, kann die allgemeine Messgleichung unter Anwendung der Mechanik starrer Körper zu

$$\boldsymbol{v}^r_l = \mathbf{C}^{b,T}_{r,l}\left(\mathbf{C}^{o,T}_b \boldsymbol{v}^o_{ob} + \left[\boldsymbol{\omega}^b_m - \boldsymbol{b}^b_g\right] \times \boldsymbol{l}^b_{br,l}\right), \tag{5.30}$$

$$= \mathbf{C}^{b,T}_{r,l}\left(\boldsymbol{v}^b_{ob} + \boldsymbol{v}^b_{\omega,l}\right) \tag{5.31}$$

aufstellt werden [178]. So kann die erwartete Radar-Eigengeschwindigkeit $\boldsymbol{v}^r$ auf Basis des Filterzustands und der Drehratenmessung $\boldsymbol{\omega}^b_m$ berechnet werden. Hierbei wird berücksichtigt, dass der Hebelarm $\boldsymbol{l}^b_{br,l}$ auf Grund der vorliegenden Winkelgeschwindigkeit $\boldsymbol{\omega}^b_m$ abzüglich des Gyroskops Offsets $\boldsymbol{b}^b_g$ auch eine Geschwindigkeitskomponente $\boldsymbol{v}^b_\omega$ induziert. Es werden zwei verschiedene Modelle evaluiert: Das Modell mit vollständigem Klon [179] sowie die Verwendung des reduzierten Klons [182]. Daraus ergeben sich für den l-ten Radarsensor die folgenden zwei Messgleichungen [182]:

$$\hat{\boldsymbol{v}}^r_{l,\text{voll}} = \hat{\mathbf{C}}^{b,T}_{r,l,\text{Klon}}\left(\hat{\mathbf{C}}^{o,T}_{b,l,\text{Klon}} \hat{\boldsymbol{v}}^o_{l,\text{Klon}} + \left[\boldsymbol{\omega}^b_m - \hat{\boldsymbol{b}}^b_{g,l,\text{Klon}}\right] \times \hat{\boldsymbol{l}}^b_{br,l,\text{Klon}}\right), \tag{5.32}$$

$$\hat{\boldsymbol{v}}^r_{l,\text{approx}} = \hat{\mathbf{C}}^{b,T}_{r,l,\text{Basis}}\left(\hat{\mathbf{C}}^{o,T}_{b,l,\text{Klon}} \hat{\boldsymbol{v}}^o_{l,\text{Klon}} + \left[\boldsymbol{\omega}^b_m - \hat{\boldsymbol{b}}^b_{g,\text{Basis}}\right] \times \hat{\boldsymbol{l}}^b_{br,l,\text{Basis}}\right). \tag{5.33}$$

Dabei sind alle Zustände des Filters mit einem Dach gekennzeichnet sowie angegeben, ob der Zustand aus dem Basiszustand (5.17) oder dem korrespondierenden Klon (5.23) stammt.

Mit Hilfe der Fehlerdefinition des Error-State-Kalman-Filters (2.1) kann das Residuum $\boldsymbol{r}_{\text{v},l}$ aus der erwarteten Radargeschwindigkeit $\hat{\boldsymbol{v}}^r_l$ und der Messung $\boldsymbol{v}^r_{m,l}$ als

$$\boldsymbol{r}_{\text{v},l} = \hat{\boldsymbol{v}}^r_{l,\{\text{voll,approx}\}} - \boldsymbol{v}^r_{l,m} \tag{5.34}$$

aufgestellt werden. Linearisierung des Residuums um den aktuellen Nominalzustand liefert die Jacobi-Matrix für den vollständigen Klon $\mathbf{H}_{\mathrm{v},l,\mathrm{voll}}$ mit

$$\mathbf{H}_{\mathrm{v},l,\mathrm{voll,ges}} = \left[\mathbf{0}_{3\times 16} \quad \mathbf{0}_{3\times 18(l-1)} \quad \mathbf{H}_{\mathrm{v},l,\mathrm{voll}} \quad \mathbf{0}_{3\times 18(L-l)}\right], \tag{5.35}$$

$$\begin{aligned}\mathbf{H}_{\mathrm{v},l,\mathrm{voll}} = \Big[\mathbf{0}_{3\times 3} \quad \mathbf{H}_{v,l,Klon} \quad \mathbf{H}_{\boldsymbol{\Psi}_{ob,l},\mathrm{Klon}} \quad \mathbf{H}_{bg,l,Klon} \; \backslash \\ \mathbf{H}_{l^b_{br,l},\mathrm{Klon}} \quad \mathbf{H}_{\boldsymbol{\Psi}_{br,l,\mathrm{Klon}}}\Big]\end{aligned} \tag{5.36}$$

$$\mathbf{H}_{v,l,\mathrm{Klon}} = \hat{\mathbf{C}}^{b,T}_{r,l,\mathrm{Klon}} \hat{\mathbf{C}}^{o,T}_{b,l,\mathrm{Klon}}, \tag{5.37}$$

$$\mathbf{H}_{\boldsymbol{\Psi}_{ob},l,\mathrm{Klon}} = \hat{\mathbf{C}}^{b,T}_{r,l,\mathrm{Klon}} \hat{\mathbf{C}}^{o,T}_{b,l,\mathrm{Klon}} \left[\hat{\boldsymbol{v}}^{o}_{l,\mathrm{Klon}}\right]\times, \tag{5.38}$$

$$\mathbf{H}_{bg,l,\mathrm{Klon}} = \hat{\mathbf{C}}^{b,T}_{r,l,\mathrm{Klon}} \left[\hat{\boldsymbol{l}}^{b}_{br,l,\mathrm{Klon}}\right]\times, \tag{5.39}$$

$$\mathbf{H}_{l^b_r,l,\mathrm{Klon}} = \hat{\mathbf{C}}^{b,T}_{r,l,\mathrm{Klon}} \left[\boldsymbol{\omega}^{b}_{m,l,\mathrm{Klon}} - \hat{\boldsymbol{b}}^{b}_{g,l,\mathrm{Klon}}\right]\times, \tag{5.40}$$

$$\begin{aligned}\mathbf{H}_{\boldsymbol{\Psi}_{br,l},\mathrm{Klon}} = \hat{\mathbf{C}}^{b,T}_{r,l,\mathrm{Klon}} \Big[\hat{\mathbf{C}}^{o,T}_{b,l,\mathrm{Klon}} \hat{\boldsymbol{v}}^{o}_{l,\mathrm{Klon}} \; \backslash \\ +[\boldsymbol{\omega}^{b}_{m} - \hat{\boldsymbol{b}}^{b}_{g,l,\mathrm{Klon}}] \times \hat{\boldsymbol{l}}^{b}_{br,l,\mathrm{Klon}}\Big]\times,\end{aligned} \tag{5.41}$$

wobei $[\square]\times$ wie in (A.2) des Anhang definiert ist. Eine detaillierte Herleitung der einzelnen Jacobi-Matrizen ist im Anhang A.3.1 gegeben. Die Jacobi-Matrizen für den approximierten Klon sind im Anhang in Abschnitt A.3.2 aufgeführt.

Als Messrauschen $\mathbf{R}_{v,l}$ wird die geschätzte Kovarianzmatrix $\mathbf{P}_{r,l}$ aus der Radar-Eigengeschwindigkeitsschätzung verwendet. Hierbei wird explizit die volle Matrix anstatt einer Diagonalmatrix verwendet, da die einzelnen Komponenten von $\mathbf{P}_{v,l}$ nicht als stochastisch unabhängig angenommen werden können. Ein Mahalanobis-Distanz Test, siehe Anhang A.1, wird final durchgeführt, um Outlier herauszufiltern. Hierfür wird die Mahalanobis Distanz ausgewertet und bei Überschreiten eines Schwellwerts wird kein Filter-Update durchgeführt.

5.4.3 Radar-Gierwinkelupdate

Wie in Doer et al. [181] diskutiert, ist der Drift des Gierratenoffsets die primäre Quelle von Positionsfehlern des Navigationssystems bei längeren Läufen. Dieser ist zwar prinzipiell beobachtbar, was sich an dem von 0 verschiedenen Ausdruck in (5.39) erkennen lässt, jedoch nicht mit der benötigten Güte. Bei einem Hebelarm von 0,1 m und einem Gierratenoffsetfehler von 0,5 °/s würde für $\boldsymbol{v}^b_\omega$ ein Geschwindigkeitsfehler von 0,87 mm/s entstehen [181]. Dies ist eine Größenordnung unterhalb der Messgenauigkeit der Radar-Eigengeschwindigkeit, daher ist eine Stützung nur mittels Radar-Eigengeschwindigkeit nicht praktikabel. Daher wurde die Gierwinkelstützung mit der Veröffentlichung Doer et al. [181] vorgeschlagen und in Doer et al. [182] für Multi-Radar-Setups erweitert.

Da die Radar-Gierwinkelschätzung, siehe Unterabschnitt 5.3.4, bereits eine Messung des globalen Gierwinkels bereitstellt, ist das Filter-Update einfach umsetzbar. Das hier eingesetzte Modell stellt dabei eine Approximation dar. Sowohl $\mathbf{C}^b_r$ der Radar-Kalibrierung als auch der Roll- und Pitch-Winkel werden bei der Transformation ins stabilisierte Koordinatensystem (5.8) als gegeben betrachtet. Da das Radar- Gierwinkelupdate erst nach erfolgreicher Initialisierung des Manhattan-Welt-Winkels γ_0 ausgeführt werden kann und die Radar Kalibrierung bis dahin in der Regel mit genügender Genauigkeit geschätzt ist, wird diese Approximation als akzeptabel eingestuft.

Das Residuum wird aufgestellt unter Verwendung der Filterdefinition (2.11):

$$r_\psi = \widehat{\psi}_{ob} - \psi_{ob,m}. \tag{5.42}$$

und die Linearisierung ergibt die Jacobi-Matrix $\mathbf{H}_\psi$ unter Annahme des l-ten von L-Radarsensoren:

$$\mathbf{H}_{\psi,\text{voll}} = \begin{bmatrix} \mathbf{0}_{1\times(16+18(l-1))} & H_{\psi^o_{b,l}} & \mathbf{0}_{(1\times 18(L-l))} \end{bmatrix}, \tag{5.43}$$

$$\mathbf{H}_{\psi,\text{approx}} = \begin{bmatrix} \mathbf{0}_{1\times 16+(9(l-1))} & H_{\psi^o_{b,l}} & \mathbf{0}_{1\times(9(L-l))} \end{bmatrix}, \tag{5.44}$$

$$H_{\psi^o_{b,l}} = 1. \tag{5.45}$$

Das Messrauschen wird fest parametrisiert und ein abschließender Mahalanobis-Distanz-Test, siehe Anhang A.1, entfernt Outlier.

5.4.4 Barometer-Update

Das Barometer-Update kann optional genutzt werden und verbessert die z-Komponente der Positionsschätzung, insbesondere auf lange Sicht. Hierzu wird zunächst der gemessene Luftdruck mit der barometrischen Höhenformel in eine relative Höhenmessung umgerechnet:

$$h_{\mathrm{baro,m}} = \frac{R\,T_0}{g_0\,M} \log\left(\frac{p_1}{P_0}\right) - h_0 \tag{5.46}$$

mit der idealen Gaskonstante $R = 8{,}314\,\mathrm{J/K/mol}$ und der mittleren molaren Masse der Atmosphäre $M = 0{,}028\,96\,\mathrm{kg/mol}$. Als Referenzhöhe wird wie üblich das Meeresniveau angenommen. Somit gilt für die Erdbeschleunigung $g_0 = 9{,}807\,\mathrm{m/s^2}$ bei einer Temperatur von $K = 288{,}15\,\mathrm{K}$ und einem Referenzluftdruck von $P_0 = 1013\,\mathrm{hPa}$.

Da lokale Schwankungen im Luftdruck mittels des Barometer Offsets b_b im Filterzustand mitgeschätzt werden, ergibt sich das Residuum zu:

$$r_{\mathrm{baro}} = \widehat{p}^{o}_{ob,z} + \widehat{b}_b - h_{\mathrm{baro,m}}. \tag{5.47}$$

mit Jacobi-Matrix $\mathbf{H}_{\mathrm{baro}}$ bei Zustandsgröße N:

$$\mathbf{H}_{\mathrm{baro}} = \begin{bmatrix} \mathbf{0}_{1\times2} & 1 & \mathbf{0}_{1\times12} & 1 & \mathbf{0}_{1\times(N-16)} \end{bmatrix}. \tag{5.48}$$

5.4.5 Ergebnisse

Im Folgenden Abschnitt wird zunächst das gesamte Navigationsfilter in Simulation und anschließend mit Realdatensätzen evaluiert. Die Realdatensätze bestehen aus getragenen und geflogenen Datensätzen und decken sowohl Innenräume als auch urbane Umgebungen ab. Abschließend wird der Einsatz der Radar-inertialen Navigation für den Einsatz zur präzisen online Navigation autonomer UAS Flüge auch in erschwerten Sichtbedingungen in Abschnitt 5.4.5 demonstriert.

Simulation

Analog zur Simulation der Radar-Eigengeschwindigkeit aus Abschnitt 5.2.5 werden Monte-Carlo Simulationen durchgeführt, um die Performance der Radar-inertialen Navigation zu evaluieren. Dabei werden die Radar-Eigengeschwindigkeitsmessungen sowie Inertialsensordaten simuliert. Es werden dabei wiederum zwei Szenarien betrachtet. Das Szenario *niedrige-Dynamik* weist eine maximale Geschwindigkeit von 2 m/s auf. Das Szenario *hohe-Dynamik*, erreicht maximale Geschwindigkeiten von 5 m/s. Zusätzlich unterscheiden sich beide Szenarien hinsichtlich der maximal auftretenden Winkelgeschwindigkeiten und Beschleunigungen, wie in Tabelle 5.4 spezifiziert. Beide Szenarien sind hinsichtlich der Zeitdauer gleich lang.

Für eine realitätsnahe Simulation werden die Messdaten de IMU mit additiven Gaußschen Messrauschen beaufschlagt. Diese orientieren sich an den Eigenschaften der realen Messdaten wie sie bei einem Flug der Flugplattform auftreten und wurden experimentell aus Testflugdaten bestimmt. Auf Grund der Vibrationen ist des Messrauschen der IMU deutlich größer als das im Datenblatt spezifizierte Messrauschen und stellt daher eine konservative Simulation dar. Das Messrauschen der Radar-Eigengeschwindigkeit wurde ebenso gemäß der beobachteten Messfehler aus den Motion-Capture Datensätzen aus Abschnitt 5.2.5 experimentell bestimmt. Dabei ist das Messrauschen der x^b-Achse (nach vorn) mit 0,06 m/s am geringsten, das der y^b-Achse (nach rechts) mit 0,09 m/s leicht größer und das der z^b-Achse (nach oben) mit 0,10 m/s am größten. Dies modelliert die experimentellen Beobachtungen. Tabelle 5.5 fasst die Varianzen des Messrauschen aller Sensordaten zusammen.

In der Simulation wird ein Setup von drei Radarsensoren, wie auch in der Multisensorplattform aus Abschnitt 3.1 verbaut, betrachtet. Neben der Güte der Positions- und Geschwindigkeitsschätzung wird auch die Genauigkeit der online Kalibrierung, also der Translation $\boldsymbol{l}_{br}^{b}$ sowie der Rotation $\boldsymbol{q}_{r}^{b}$ zwischen dem jeweiligen Radarsensor und dem Bodyframe analysiert. Für jeden Monte-

Kenngröße	Szenario *niedrige-Dynamik*	Szenario *hohe-Dynamik*
$\lvert\boldsymbol{v}_{\max}\rvert$	2 m/s	5 m/s
$\lvert\boldsymbol{\omega}_{\max}\rvert$	50 °/s	200 °/s

Tabelle 5.4: Wichtige Kenngrößen der beiden Simulationsszenarien

Carlo Simulationsdurchlauf wird daher auch die extrinsische Kalibrierung aus einer mittelwertfreien Normalverteilung mit Varianzen, wie in Tabelle 5.5 spezifiziert, neu gezogen. Somit ergibt sich für jeden Monte-Carlo Lauf eine neue Konfiguration der extrinsischen Kalibrierung der Radarsensoren. Zusätzlich wird das Messrauschen für jeden Durchlauf neu erstellt und der initiale Offset des Beschleunigungssensors gemäß Tabelle 5.5 bestimmt. Es wurden 100 Monte-Carlo-Durchläufe ausgeführt, um eine verlässliche Aussage hinsichtlich der Performance zu ermöglichen. Für jeden Monte-Carlo Lauf wurde der Fehler mit Hilfe der Referenz bestimmt und wird im Folgenden analysiert.

Abbildung 5.11 zeigt die Ergebnisse für die Fehler der Position, Geschwindigkeit, Lage, Offset des Beschleunigungssensors sowie des Drehratensensors für alle Monte-Carlo Läufe. Der $\pm 3\sigma$-Schlauch der geschätzten Varianzen ist dabei in schwarz gestrichelt eingetragen. Es liegen alle Monte-Carlo Läufe innerhalb des $\pm 3\sigma$-Schlauchs, was auf eine konsistente Schätzung des Navigationsfilters hindeutet. Die Verläufe der Unsicherheit der Position sowie des Gierwinkels zeigen einen wachsenden Verlauf. Die Unsicherheiten der Lage in Roll und Pitch sowie der Geschwindigkeit konvergieren schnell. Die Verläufe der Unsicherheit der IMU-Offsets konvergieren langsam. Die initialen Offsets der Drehrate weisen einen für jeden Lauf verschiedenen Fehler auf, da dieser anhand der vorliegenden Messungen jeweils durch Mittelung initialisiert wird.

Am Verlauf der geschätzten Kovarianzen lässt sich das Systemverhalten verifizieren. Da es sich um einen Odometrie-Ansatz handelt, ist die Position sowie der Gierwinkel nicht global beobachtbar. Die korrespondierende Unsicherheit muss daher mit der Zeit anwachsen. Die Lagewinkel in Roll und Pitch sind dagegen auf Grund der Erdbeschleunigung global beobachtbar und müssen daher konvergieren. Durch die Stützinformation der Radar-Eigengeschwindig-

Kenngröße	Wert
Messrauschen Beschleunigung	$0{,}4\,\mathrm{m/s^2}$
Messrauschen Winkelgeschwindigkeit	$2{,}5\,°/\mathrm{s}$
Messrauschen Radar-Eigengeschwindigkeit (xyz)	$[0{,}06,\ 0{,}09,\ 0{,}10]\mathrm{m/s}$
Initiale Unsicherheit $\boldsymbol{b}_a$	$0{,}1\,\mathrm{m/s^2}$
Initiale Unsicherheit $\boldsymbol{l}_{br}^{b}$	$0{,}02\,\mathrm{cm}$
Initiale Unsicherheit $\boldsymbol{\Psi}_{br}$	$1{,}5°$

Tabelle 5.5: Spezifizierung des Messrauschens der Simulation

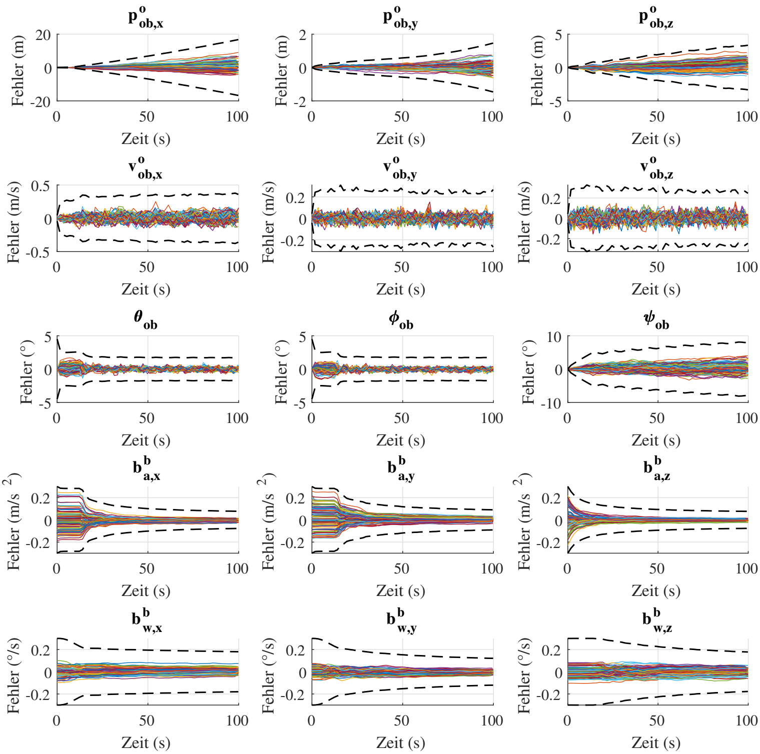
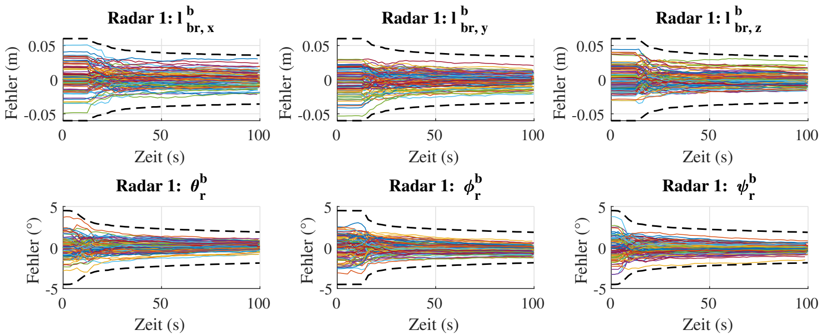
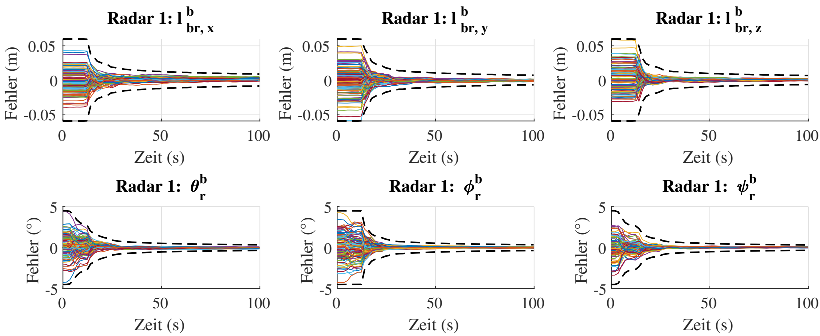


Abbildung 5.11: Simulationsergebnis für das Szenario *niedrige-Dynamik* für 100 Monte-Carlo Läufe. Der $\pm 3\sigma$-Schlauch der geschätzten Unsicherheit ist schwarz gestrichelt eingezeichnet. Alle Monte-Carlo Läufe liegen innerhalb des $\pm 3\sigma$-Schlauchs, was auf eine konsistente Schätzung hindeutet.

keit, wird die Geschwindigkeit auch global beobachtbar und muss folglich auch konvergieren. Ferner, werden die Offsets des Beschleunigungssensors sowie des Drehratensensors durch die Stützinformation beobachtbar und müssen folglich auch konvergieren. Die Erwartungen decken sich mit dem Simulationsergebnis aus Abbildung 5.11. Es zeigt sich jedoch auch, dass die Schätzung des Drehratenoffsets nur mit geringer Güte möglich ist und etwa bei 0,2 °/s konvergiert. Dies führt zum Anwachsen der Gierwinkel-Unsicherheit und schließlich zu einem Drift der Gierwinkelschätzung der reinen Radar-inertialen Navigation



(a) Simulationsszenario niedrige Dynamik



(b) Simulationsszenario hohe Dynamik

Abbildung 5.12: Ergebnis der Monte-Carlo Simulation für die Schätzung der extrinsischen Kalibrierung der Translation $\boldsymbol{l}^b_{br}$ und der Rotation $\boldsymbol{q}^b_r$ bei einem Setup von drei Radarsensoren. Der geschätzte $\pm 3\sigma$ Schlauch ist schwarz gestrichelt. Die einzelnen Monte-Carlo Simulationen liegen innerhalb dieses Schlauchs, was auf eine konsistente Schätzung hindeutet. Die Schätzung ist deutlich genauer bei dem Szenario mit *hoher Dynamik* (unten) als bei *niedriger Dynamik* (oben).

ohne Radar-Gierwinkelstützung. Es ist daher sehr wichtig, den Drehratenoffset exakt zu kalibrieren, um den Drift des Gierwinkels möglichst gering zu halten.

Abbildung 5.12 zeigt die Verläufe der extrinsischen Kalibrierung für beide Simulationsszenarien. Dabei sind exemplarisch die Verläufe des 1. Radarsensors gezeigt, die Verläufe aller drei Radarsensoren sind im Anhang in Abbildung A.1 und A.2 dargestellt. Alle Monte-Carlo Läufe liegen wiederum innerhalb der geschätzten $\pm 3\sigma$-Schläuche, was auch auf eine konsistente

Schätzung des extrinsischen Kalibrierzustände schließen lässt. Beide Szenarien erzielen jedoch einen deutlichen Unterschied hinsichtlich der Güte der extrinsischen online Kalibrierung. Die initiale Unsicherheit der Zustände ist zwar bei beiden Szenarien gleich, konvergiert jedoch für den translatorischen Teil beim Szenario *niedrige-Dynamik* gegen 33 mm im Gegensatz zu 6 mm im Szenario *hohe-Dynamik*. Ähnliches Konvergenz-Verhalten ergibt sich auch für den rotatorischen Teil. So konvergiert das Szenario *niedrige-Dynamik* bei 1,88° und *hohe-Dynamik* bei 0,34°. Dieses Verhalten spiegelt das Systemverhalten der Online Kalibrierung wider. Die korrespondierende Jacobi-Matrix der Translation (5.40) besteht aus einem Kreuzprodukt mit der vorliegenden Winkelgeschwindigkeit $\boldsymbol{\omega}_m^b$. Folglich koppeln Fehler in $\boldsymbol{l}_{br}^b$ bei höheren Winkelgeschwindigkeiten stärker ein, was zu einer verbesserten Schätzung dieser Größe führt. Da das Szenario *hohe-Dynamik* mit einer maximalen Winkelgeschwindigkeit von 200 °/s einen viermal größeren Wert aufweist, erklärt sich die stark verbesserte Schätzung der translatorischen Transformation. Die Jacobi-Matrix des rotatorischen Anteils (5.41) zeigt eine starke Abhängigkeit von der vorliegenden Geschwindigkeit $\boldsymbol{v}_{ob}^o$. Somit ergibt sich eine bessere Schätzung bei höherer Geschwindigkeit, weil so Fehler der Rotation stärker einkoppeln. Durch die höhere Geschwindigkeit beim Szenario *hohe-Dynamik* (5 m/s vs. 2 m/s) ergibt sich die verbesserte Schätzung der rotatorischen Transformation.

Indoor-Szenarien

Im Folgenden erfolgt die umfassende Evaluation der Radar-inertialen Navigation in Innenräumen. Dabei werden verschiedene Umgebungen und Konfigurationen der Radar-inertialen Navigation evaluiert. Es werden vier Datensätze betrachtet, wie in Abbildung 5.13 dargestellt. Der Datensatz *Office-Floor* deckt dabei das ganze Stockwerk eines möblierten Bürogebäudes ab, einschließlich eines Ganges, Büros und Computerräume. Ein gesamtes Stockwerk bestehend aus mehreren Laboren schließt der *Lab-Floor* Datensatz ein. Lange Gänge mit wenig Struktur durch Möblierung zeichnen den Datensatz *Basement* aus. Eine große Werkstatt mit einer Vielzahl an Maschinen ist die Umgebung des letzten Datensatzes *Workshop*. Bei diesen Datensätzen handelt es sich um getragene Datensätze. Neben den Radar-inertial Sensordaten wurden auch die Daten der Intel-Realsense-Kamera, siehe Tabelle 3.1, aufgezeichnet. Diese dienen zum

Erstellen einer Pseudo-Referenz und zur Evaluation gegenüber einem visuell-inertialen Odometrie (VIO) Ansatz.

Die Pseudo-Referenz wurde mit Hilfe eines Stereo visuell-inertialen SLAMs (Simultaneous Localization and Mapping) erstellt. Bei der Erstellung der Datensätze wurde auf regelmäßige Schleifenschlüsse geachtet, sodass der SLAM sehr akkurate Ergebnisse der Trajektorie liefert. Hierfür wurde VINS [110] eingesetzt, wobei Schleifenschlüsse aktiviert wurden damit so aus dem VIO-Ansatz ein SLAM-Ansatz wird. Die erstellten Trajektorien wurden anschließend mittels des Grundrisses und der geschickten Wahl des gelaufenen Pfades, wie zum Beispiel dem Durchschreiten der Türen exakt in der Mitte, manuell hinsichtlich der Skalenfehler korrigiert. Die so erstellte Pseudo-Referenz ist sehr genau, wie in der Darstellung mit dem Grundriss in Abbildung 5.13 zu erkennen ist. Bei allen Datensätzen wurden mindestens zwei Runden des annähernd gleichen Pfades gelaufen. Obwohl die Radar-inertiale Navigation auch in schlechten Sichtbedingungen, wie Dunkelheit oder Nebel eingesetzt werden kann, weisen alle Datensätze gute Sichtbedingungen auf. So kann die Pseudo-Referenz erstellt als auch der Vergleich zu VIO durchgeführt werden.

Abbildung 5.14 zeigt das Ergebnis des Datensatzes *Office-Floor* visualisiert auf dem Grundriss. Dabei ist der Pfad der geschätzten Position der Radar-inertialen Navigation (x-RIO) in blau und die Pseudo-Referenz in rot eingezeichnet. Die Radar-inertiale Navigation wurde dabei in der Konfiguration mit drei Radarsensoren sowie Gierwinkel- und Barometerstützung eingesetzt. Der Pfad von x-RIO weist nur eine sehr kleine Positionsdrift auf. Der finale Positionsfehler beträgt lediglich 0,24 m. Bezogen auf die Trajektorienlänge ergibt sich daher ein relativer finaler Positionsfehler von 0,06 %. Da eine Referenz für die Datensätze vorliegt, kann Evaluation der ATE, siehe Abschnitt 2.5 im Grundlagenkapitel, eingesetzt werden. Der ATE ist der translatorische RMS-Fehler aller Zeitpunkten, an denen eine Referenz vorliegt und ist daher eine sehr oft eingesetzte Metrik [159]. Die Berechnung des ATE wurde mit der Toolbox *rpq_trajectory_evaluation* [159] durchgeführt.

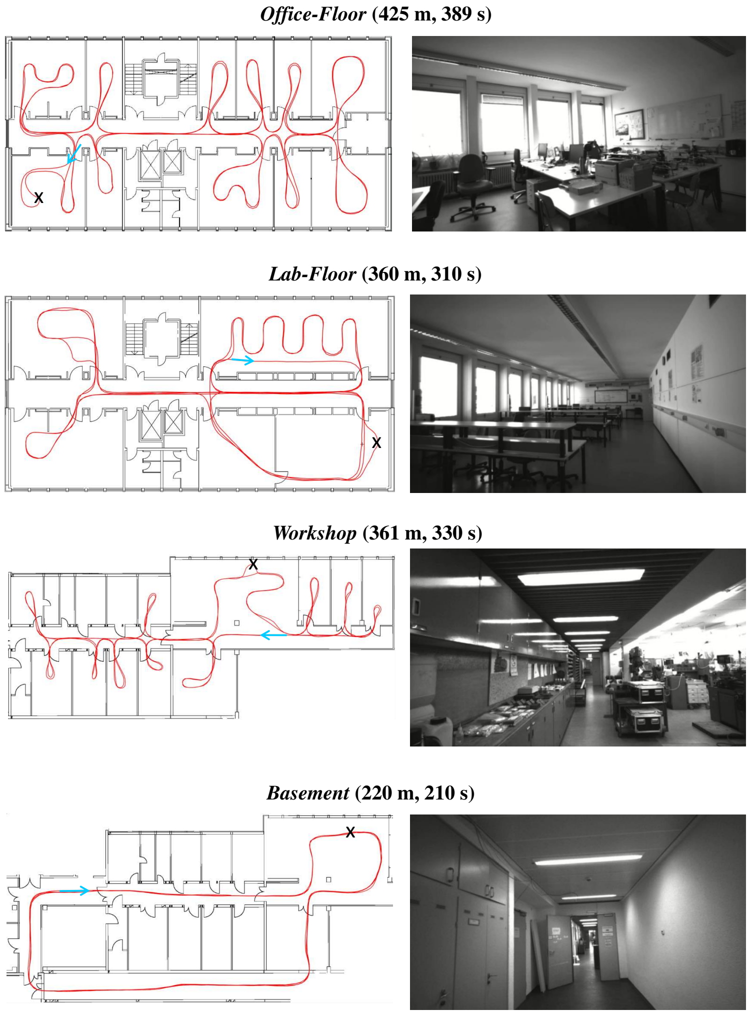


Abbildung 5.13: Übersicht der vier Indoor-Datensätze mit der Pseudoreferenz und Grundriss (links) und Beispielbildern (rechts). Die blauen Pfeile markieren die Position der Multisensorplattform bei dem jeweiligen Beispielbild. Die Start- und Endposition ist jeweils durch das schwarze Kreuz markiert. Es wurden jeweils mehrere Runden des gleichen Pfades absolviert.

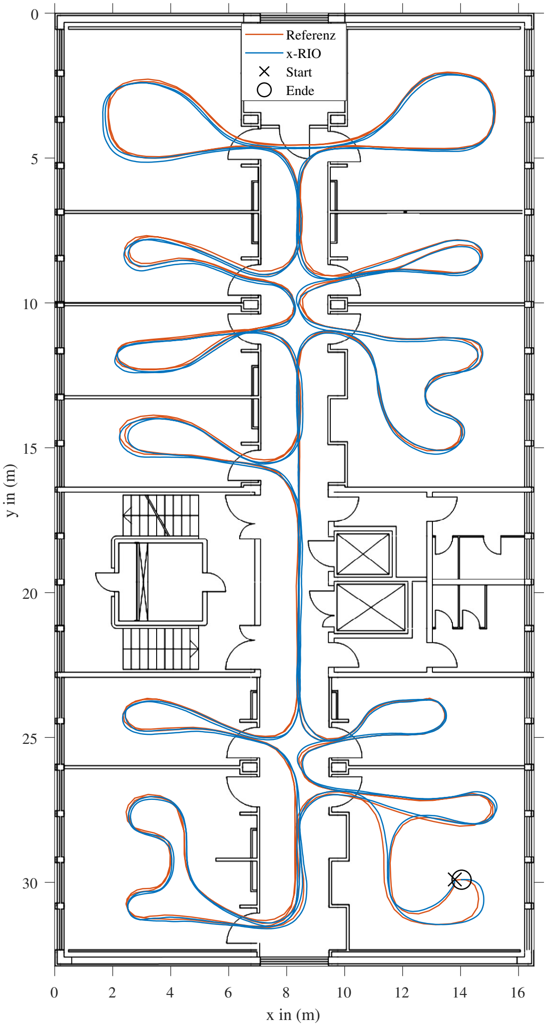


Abbildung 5.14: Geschätzter Pfad (blau) und Pseudoreferenz (rot) mit Grundriss des Datensatzes *Office-Floor* nach Doer et al. [182]. Die Start- (Kreuz) und Endposition (Kreis) sind gleich. Es wurden zwei Runden des approximativ gleichen Pfades gelaufen. x-RIO folgt dabei der Referenz sehr genau und weist nur eine kleine Positionsfehlerdrift auf. Der finale Positionsfehler beträgt 0,24 m, was 0,06 % der Trajektorienlänge entspricht.

Diese Art der Evaluation wurde für die folgenden Ansätze ausgeführt:

- x-RIO: Das entwickelte Radar-inertiale Navigationsfilter, wobei verschiedene Konfigurationen hinsichtlich der Anzahl an Radarsensoren und der Art der Stützung evaluiert werden. Die Radarsensoren wurden mittels TDMA angesteuert zur Vermeidung gegenseitiger Störung.
- VINS [110]: Stereo visuell-inertiale Odometrie (VIO) Ansatz, nach [33] einer der genauesten Open-Source VIO-Ansätze.

Tabelle 5.6 zeigt die Ergebnisse für alle vier Datensätze für verschiedene Konfigurationen von x-RIO bezüglich der Radarsensoren sowie Stereo-VINS. Alle x-RIO Konfigurationen wurden mit aktivierter Gierwinkel- und Barometerstützung prozessiert. Für jeden Datensatz wurde der ATE berechnet und in der letzten Spalte von Tabelle 5.6 der Mittelwert aller Läufe des jeweiligen Ansatzes angegeben. x-RIO erzielt mit allen Radarsensorkonfiguration bessere Ergebnisse als Stereo-VINS außer dem mittleren Radarsensor im Schnitt über alle Datensätze. Während x-RIO mit dem linken und rechten Sensor ähnliche Ergebnisse erzielt (0,30 m vs. 0,27 m), erreicht x-RIO mit dem mittleren Radarsensor das schlechteste Ergebnis mit 0,44 m. Wie in Tabelle 3.1 aufgeführt, besitzt der mittlere Radarsensor (RISK) einen Sichtbereich von 120° in Azimut und 30° in Elevation. Die Winkelauflösung, also die minimale Winkeldifferenz, um bei gleicher Entfernung und Relativgeschwindigkeit zwei Objekte trennen zu können, liegt bei 14° in Azimut und 58° in Elevation. Der linke und rechte Radarsensor (AOP) dagegen haben einen Sichtbereich von 120° und

Algorithmus	Radarsensor	*Office-Floor*	*Lab-Floor*	*Basement*	*Workshop*	Mittelwert
	linker (l)	0,16	0,31	0,51	0,23	0,30
	rechter (r)	0,20	0,32	0,24	0,30	0,27
x-RIO	mittlerer (m)	0,25	0,44	0,66	0,40	0,44
	l und r	0,18	0,26	0,36	0,19	0,25
	l, m und r	0,13	0,23	0,38	0,20	0,23
Stereo-VINS		0,48	0,47	0,52	0,18	0,41

Tabelle 5.6: Vergleich von x-RIO in verschiedenen Konfigurationen hinsichtlich der Radarsensoren und Stereo-VINS nach Doer et al. [182]. Bei allen x-RIO Läufen war sowohl die Gierwinkel- als auch die Barometerstützung aktiviert. Die Tabelle zeigt den ATE in m. Alle x-RIO Konfigurationen, außer der mittlere Sensor, erreichen einen besseren mittleren ATE als Stereo-VINS.

	Radarsensor				
x-RIO-Konfiguration	l	r	m	l-r	l-m-r
Nur Radar-Eigengeschwindigkeit	0,46	0,44	1,07	0,47	0,44
+ Barometerstützung	0,39	0,30	0,42	0,37	0,32
+ Gierwinkelstützung	0,30	0,26	0,42	0,24	0,24
+ approximierter Klon	0,30	0,27	0,44	0,25	0,23

Tabelle 5.7: Analyse der verschiedenen x-RIO- und Radarsensorkonfigurationen. Für jede Konfiguration wurden alle vier Indoor-Datensätze prozessiert und der mittlere ATE in m berechnet.

eine Winkelauflösung von 29° in Azimut und Elevation. Der größere Sichtbereich in Elevation und die bessere Winkelauflösung führen zu einer besseren Schätzung der z-Komponente der Geschwindigkeit und folglich der Position und damit zu einem niedrigeren Fehler.

Tabelle 5.7 präsentiert die Analyse von x-RIO bezüglich der Filter-Konfiguration und Anzahl an Radarsensoren. Dabei ist jeweils der ATE-Mittelwert aller vier Datensätze angegeben. Es zeigt sich bei allen Radarsensor-Konfigurationen eine deutliche Verbesserung des ATE unter Verwendung der Stützung. Dabei ist bei der Verwendung des linken oder des rechten Radarsensors die gleiche Charakteristik zu beobachten. Die Konfiguration mit dem mittleren Radarsensor profitiert sehr stark von der Barometerstützung, was eine Verbesserung des ATE von 60 % zur Folge hat. Die Nutzung des linken und rechten Radarsensors mit Barometer- und Gierwinkelstützung erzielt bessere Ergebnisse als die Nutzung von nur einem Sensor mit voller Stützung. So wird der ATE mit zusätzlicher Stützung, um 17 % im Vergleich zum Setup mit nur dem linken Radarsensor verbessert. Die Gierwinkelstützung zeigt vor allem beim Einsatz des linken und/oder rechten Radarsensors eine erhebliche Verbesserung des ATE. Die stärkste Verbesserung ist dabei bei dem Setup mit zwei Radarsensoren (l-r: 35 %) sowie drei Radarsensoren (l-m-r: 25 %) zu beobachten.

Die Verwendung von allen drei Radarsensoren hat in den betrachteten Szenarien gegenüber zwei Radarsensoren zu keiner deutlichen Verbesserung geführt. Ein solches Setup könnte jedoch vorteilhaft sein, wenn auf Grund von mangelnder Struktur innerhalb des Sichtbereichs eines der Radarsensoren sich im Scan keine Objekte befinden. Dies ist jedoch in Innenräumen auf Grund der maximalen Reichweite der eingesetzten Radarsensorik von 50 m in den betrachteten Szenarios nicht aufgetreten. Die Konfiguration mit dem reduzierten

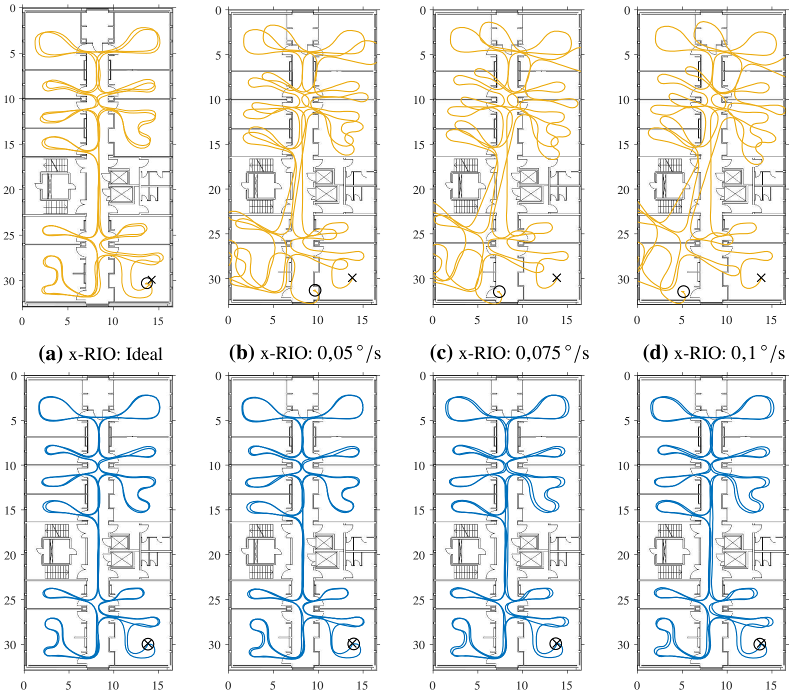
Klon erreicht annähernd die gleichen Ergebnisse und zeigt somit, dass die Annahmen aus Abschnitt 5.4.2 hinsichtlich des approximierten Klons zulässig sind.

Die Nutzung des approximierten Klons ist vorteilhaft bezüglich der benötigen Rechenressourcen. Tabelle 5.8 führt die Echtzeitfaktoren aller Konfiguration aus Tabelle 5.7 auf. Der Echtzeitfaktor ist die Datensatzlänge geteilt durch die Ausführungszeit. Ein Faktor größer eins bedeutet daher, dass der Algorithmus schneller als Echtzeit ausführbar ist. Tabelle 5.8 präsentiert die Echtzeitfaktoren des Intel-NUC-i7 sowie in Klammern die des *Up-Cores*. Es zeigt sich in Tabelle 5.8, dass x-RIO ohne Stützung die höchsten Echtzeitfaktoren erreicht. So erzielt x-RIO mit nur einem Radarsensor Echtzeitfaktoren von 157 auf dem Intel-NUC-i7 und 24 auf dem *Up-Core*. Auch bei Verwendung von zwei Radarsensoren und allen Stützinformationen erreicht die Implementierung des reduzierten Klons 77-fache Echtzeit auf dem *Intel-NUC-i7* und 12-fache Echtzeit auf dem *Up-Core*. Die Nutzung des reduzierten Klons ergibt im Schnitt eine Verbesserung des Echtzeitfaktors von 10 %. Da der reduzierte Klon annähernd die gleichen Ergebnisse wie der vollständige Klon erzielt, siehe Tabelle 5.7, wird dieser im Folgenden verwendet.

Abbildung 5.15 veranschaulicht die Ergebnisse der Gierwinkelstützung (x-RIO-Yaw). Der Datensatz *Office-Floor* wurde mit x-RIO und x-RIO-Yaw unter Verwendung aller drei Radarsensoren verarbeitet. Dabei wurde der initiale Gierratenoffset erst mit der idealen Kalibrierung und anschließend schrittweise verfälscht mit beiden Ansätzen prozessiert. So wird eine suboptimale Kalibrierung des Gierratenoffsets sowie ein sich ändernder Gierratenoffset,

	Anzahl an Radarsensoren		
x-RIO Konfiguration	1	2	3
Nur Radar-Eigengeschwindigkeit	157 (24)	88 (13)	56 (8)
+ Gierwinkelstützung	130 (22)	75 (11)	52 (7)
+ Barometerstützung	124 (19)	68 (10)	45 (6)
+ approximierter Klon	128 (21)	77 (12)	49 (7)

Tabelle 5.8: Echtzeitfaktoren in Abhängigkeit von x-RIO Konfigurationen und Anzahl an Radarsensoren. Der Echtzeitfaktor ist die benötige Zeit, um den Datensatz zu verarbeiten geteilt durch die Datensatzlänge. Die Ergebnisse sind für den *Intel-NUC-i7* und den *Up-Core* (in Klammern) angegeben. x-RIO erreicht Ausführungszeiten um ein vielfaches schneller als Echtzeit selbst auf dem *Up-Core*.



(a) x-RIO: Ideal **(b)** x-RIO: 0,05 °/s **(c)** x-RIO: 0,075 °/s **(d)** x-RIO: 0,1 °/s

(e) x-RIO-Yaw: Ideal **(f)** x-RIO-Yaw: 0,05 °/s **(g)** x-RIO-Yaw: 0,075 °/s **(h)** x-RIO-Yaw: 0,1 °/s

Abbildung 5.15: Vergleich von x-RIO und x-RIO-Yaw mit drei Radarsensoren für verschiedene Gierratenoffsetfehler. Der jeweils gleiche Datensatz wurde mit x-RIO (oben) und x-RIO-Yaw (unten) prozessiert. Der Gierratenoffset wurde von 0,05° bis 0,1 °/s schrittweise gegenüber der initialen Kalibrierung verfälscht. Die Gierratenstützung (x-RIO-Yaw) kann selbst Gierratenoffsetfehler von 0,1 °/s kompensieren.

wie er typischerweise bei low-cost IMUs auftritt, simuliert. Es zeigt sich bei x-RIO ein wachsender Drift des Gierwinkels, der mit dem Gierratenoffsetfehler zunimmt. Selbst ein Gierratenoffsetfehler von 0,05 °/s, siehe Abbildung 5.15b, führt sehr schnell zu einer Akkumulation des Gierwinkel- und daraus resultierend des Positionsfehlers. Zwar ist der Gierwinkel beobachtbar, jedoch nicht mit einer ausreichenden Genauigkeit, wie auch simulativ in Abschnitt 5.4.5 diskutiert. Je größer der Gierwinkelfehler, desto größer auch der finale Positionsfehler. Die zeigt sich in Abbildung 5.15 deutlich, da die x-Markierung

(Start) und o-Markierung (Ende) übereinander liegen müssen. Die Gierwinkelstützung kann effektiv die akkumulierenden Fehler kompensieren. So zeigt sich bei x-RIO-Yaw in Abbildung 5.15f - 5.15h trotz des anwachsenden initialen Gierratenoffsetfehlers ein sehr ähnliches Ergebnis der geschätzten Trajektorie. Die Stützung mittels Manhattan-Welt Annahmen ist also ein effektiver Weg zur Stützung des Gierwinkels auch bei großen initialen Kalibrierfehlern oder bei einer Änderung des Gierwinkeloffsets während des Laufs.

Outdoor-Szenarien

In diesem Abschnitt erfolgt die Evaluation der Radar-inertialen Navigation auf den *Campus-Loops*-Datensätzen. Diese wurden in urbanen Gebieten auf dem Campus des KIT aufgezeichnet. Aus rechtlichen Gründen konnten keine Flüge durchgeführt werden, weshalb es sich auch hierbei um getragene Datensätze handelt. Die aufgezeichneten Datensätze bilden typische urbane Szenarien einschließlich Gebäuden und Vegetation ab. Alle vier Datensätze sind in Abbildung 5.16 dargestellt wobei die eingezeichneten Pfade die Lösung von x-RIO sind.

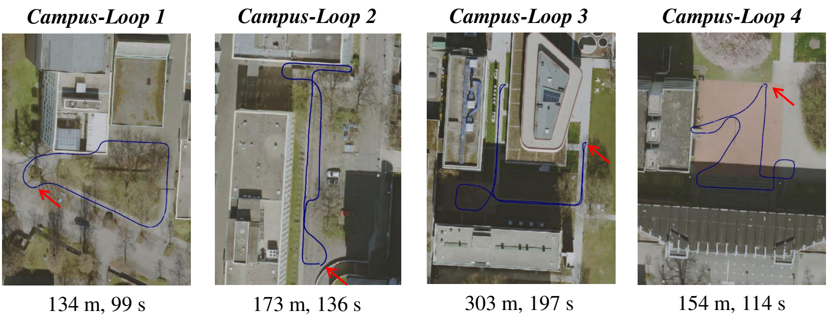


Abbildung 5.16: Luftbild der *Campus-Loops*-Datensätze mit dem Ergebnis von x-RIO des linken Radarsensors mit Gierwinkel- und Barometerstützung (blau) nach Doer et al. [172]. Start- und Endposition waren gleich für alle Läufe und sind durch den roten Pfeil in den Bildern gekennzeichnet. Luftbild: © Stadt Karlsruhe.

Da weder mit monokularem noch Stereo visuell-inertialen SLAM-Ansätzen wie VINS [110] (mit Schleifenschluss) oder ORB-SLAM3 [18] eine Referenz mit ausreichender Genauigkeit erstellt werden konnte, kann eine Evaluation wie bei den Indoor-Datensätzen nicht durchgeführt werden. Die *Campus-Loops*-Datensätze weisen jedoch alle die gleichen Start- und Endpositionen auf, was eine Evaluation des relativen finalen Positionsfehlers ermöglicht. Dieser ergibt sich aus dem finalen Positionsfehlerbetrag geteilt durch die Trajektorienlänge.

Tabelle 5.9 zeigt die Ergebnisse aller vier Datensätze für die Radar-inertiale Navigation ohne (x-RIO) und mit Gierwinkel- und Barometerstützung (x-RIO-Yaw-Baro) für verschiedene Radarsensor-Kombinationen sowie das Ergebnis von VINS [110] als Vergleich zu VIO. Wie bereits bei den Indoor-Datensätzen erzielt die Radar-inertiale Navigation mit dem linken oder dem rechten Radarsensor ähnliche Ergebnisse sowohl ohne Stützung (0,73 % und 0,85 %) als auch mit Stützung (0,51 % und 0,51 %) für den mittleren Fehler. Der mittlere Radarsensor profitiert stark von der Stützung, so wird der mittlere Fehler bei Stützung von 2,13 % auf 0,73 % reduziert. Die deutlich schlechtere Genauigkeit des mittleren Radarsensors ist wieder mit dem reduzierten Sichtbereich in Elevation begründet. Die Kombination des linken und rechten Radarsensors erzielt bei den vorliegenden Datensätzen ähnliche Ergebnisse. Die Kombination aller drei Radarsensoren zeigt im Vergleich zur Kombination links-rechts eine leichte Verschlechterung. Bei den betrachteten Datensätzen war viel Struktur, zum Beispiel durch Gebäude, in den Sichtbereichen aller Radarsensoren. Diese weisen wie in Tabelle 3.2 eine maximale Reichweite von circa 50 m auf. Eine Kombination von mehreren Radarsensoren scheint dennoch vorteilhaft im Hin-

Datensatz	**x-RIO**					**x-RIO-Yaw-Baro**					**VINS**
	l	r	m	l-r	l-m-r	l	r	m	l-r	l-m-r	[110]
Campus-Loop 1	1,33	1,57	1,43	1,25	0,22	0,54	0,47	0,35	0,63	0,60	0,84
Campus-Loop 2	0,38	0,30	0,30	0,69	1,12	0,43	0,82	0,74	0,77	0,90	0,33
Campus-Loop 3	0,33	0,76	0,49	0,82	1,36	0,21	0,29	0,50	0,29	0,39	0,35
Campus-Loop 4	0,88	0,78	0,64	0,75	5,80	0,82	0,44	1,31	0,57	0,74	1,96
∅ rel. Fehler (%)	0,73	0,85	2,13	0,72	0,88	0,50	0,51	0,73	0,57	0,66	0,87

Tabelle 5.9: Ergebnis der *Campus-Loops-Datensätze* für x-RIO und visuell-inertialer Odometrie (VINS). Als Evaluationsmetrik wird der relative finale Positionsfehler (%), also die finale Positionsfehlernorm geteilt durch die Trajektorienlänge verwendet.

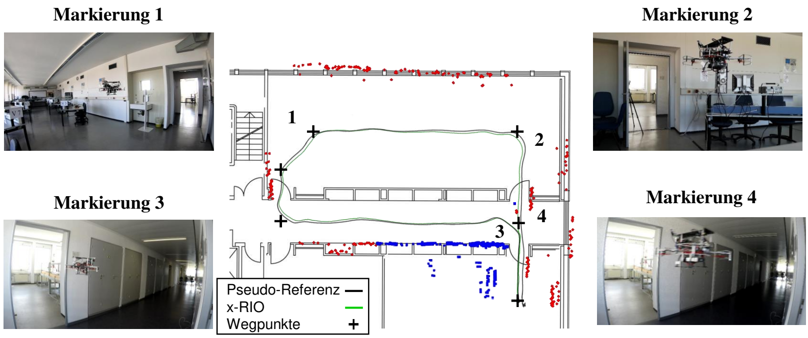
blick auf Szenarien mit asymmetrischen Szenen hinsichtlich der vorhandenen Struktur. So sind Szenarien denkbar, in denen Struktur nur auf einer Seite des UAS vorhanden ist. Während die Verwendung eines einzelnen Radarsensors hier eine robuste Navigation nicht garantieren kann, ist dies bei Verwendung mehrerer Sensoren gewährleistet. Der Vergleich zu VINS zeigt, dass die Radar-inertiale Navigation auch in Outdoor-Szenarien vergleichbare Ergebnisse bei Deaktivierung der zusätzlichen Stützung unter Verwendung des linken, rechten oder linken und rechten Radarsensors erreicht (0,73 %, 0,85 % und 0,72 % versus 0,87 %). Bei Aktivierung der Gierwinkel- und Barometerstützung erreichen alle x-RIO-Konfigurationen deutlich bessere Ergebnisse. Bei Einsatz des linken und/oder des rechten Radarsensors ergibt sich eine Verbesserung von mehr als 34 % im Vergleich zu VINS.

Demonstration autonomer Flüge

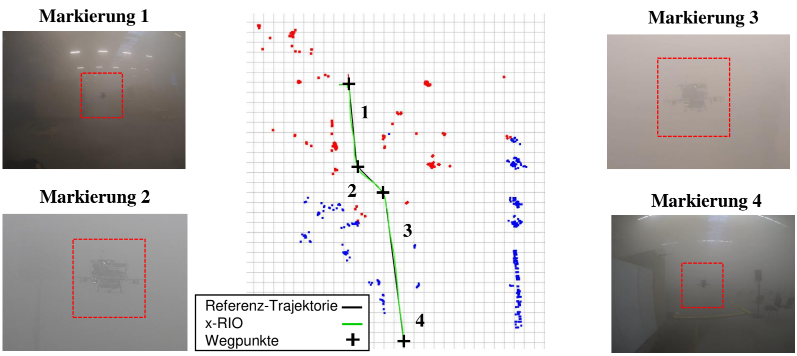
In diesem Abschnitt wird die Verwendung von x-RIO zur online Navigation eines UAS für autonome Flüge demonstriert. Dabei wird jeweils ein Trajektorienflug durchgeführt. Hierfür wird zunächst eine Ziel-Trajektorie erzeugt und diese anschließend abgeflogen, wobei ausschließlich mit x-RIO navigiert wird. Es werden zwei Szenarien demonstriert: ein Flug in einem Bürogebäude in guten Sichtbedingungen und ein Flug in dichtem Nebel am Deutschen-Rettungsrobotik-Zentrum (DRZ) bei schlechten Sichtbedingungen.

Die Erzeugung der Trajektorien erfolgt mit Hilfe von zuvor definierten Wegpunkten. Auf Basis dieser wird mit Hilfe von [100] eine glatte Trajektorie unter Minimierung des Rucks erzeugt. Jeder Zeitpunkt der Trajektorie beinhaltet die Stellgröße für die Ziel-Position, -Geschwindigkeit, -Beschleunigung, -Gierwinkel und -Gierrate. Diese werden als Stellgrößen mit einer Rate von 20 Hz an den Pixhawk-Flugregler gesendet. Durch die umfangreichen Stellgrößen können die Trajektorien auch bei dynamischen Manövern sehr genau abgeflogen werden.

Der autonome Indoor Flug in guten Sichtbedingungen ist in Abbildung 5.17a dargestellt. Die Wegpunkte, die zur Erstellung der Trajektorie genutzt wurden, sind durch die schwarzen Kreuze gekennzeichnet. Das Szenario beinhaltet insgesamt vier Türdurchflüge. Da die Türen eine Breite von 1 m aufweisen und die Flugplattform eine Kantenlänge von 45 cm hat, bleibt bei den Türdurch-



(a) Autonomer Indoor-Flug nach Doer et al. [181]



(b) Autonomer Flug in dichtem Nebel

Abbildung 5.17: Autonome Testflüge mit online Navigation durch x-RIO für Szenarien mit guten Sichtbedingungen (oben) und schlechten Sichtbedingungen (unten). Es wurden jeweils Wegpunktflüge ausgeführt, diese wurden vor dem Flug bestimmt und sind durch die schwarzen Kreuze visualisiert. x-RIO kann zur online Navigation von UAS eingesetzt werden und ermöglicht Flüge sogar in dichtem Nebel. Das rote Quadrat markiert die Flugplattform im dichten Nebel.

flügen ein Spielraum von nur 27,5 cm. In diesem Szenario wurde bewusst auf eine Kollisionsvermeidung verzichtet. So kann die hohe Güte der Navigation demonstriert werden, da Lokalisierungsfehler von über 27,5 cm zu einer Kollision führen würden. Für das Flugexperiment wurde die initiale Position und Ausrichtung der Flugplattform exakt bestimmt und dann der Trajektorienflug gestartet. Die geschätzte Trajektorie von x-RIO ist in Abbildung 5.17a in grün

dargestellt. Bei diesem Flug wurde x-RIO nur mit dem linken Radarsensor eingesetzt. Zusätzlich wurde auch eine Pseudoreferenz, wie auch bei den Indoor-Datensätzen mit Hilfe von VINS [110] mit Schleifenschlüssen, erstellt und zusätzlich in schwarz eingezeichnet. Es zeigt sich nur eine sehr kleine Drift des Positionsfehlers. Des Weiteren ist die Radar-Gierwinkelstützung visualisiert. Die blau eingefärbten Radarscan-Punkte wurden dabei zur Initialisierung verwendet und die roten Radarscan-Punkte für die Gierwinkelmessungen. Es zeigt sich, dass Manhattan-Welt-Strukturen sehr gut detektiert werden, da die Punkte entweder auf Wänden liegen oder zu Wänden ausgerichtetes Mobiliar abbilden. Der erfolgreiche Flug demonstriert dabei die Anwendbarkeit zur Online Navigation von UAS mit sehr hoher Güte auch in engen Indoor-Umgebungen.

Das zweite Szenario ist in Abbildung 5.17b dargestellt und demonstriert die Navigation in dichtem Nebel. Für diese Demonstration wurde die DRZ-Halle mit Hilfe leistungsstarker Nebelmaschinen vernebelt. Analog zur vorherigen Demonstration wurde aus Wegpunkten eine Trajektorie unter Minimierung des Rucks [100] erstellt und abgeflogen. Trotz der schlechten Sichtbedingungen mit Sichtweiten unter 1 m konnte der Flug erfolgreich durchgeführt werden. Die Radar-inertiale Navigation stellt somit eine bewiesene Möglichkeit zur Navigation von autonomen UAS auch unter schwierigen Bedingungen dar.

5.5 Fazit

In diesem Kapitel wurde zunächst die Radar-Eigengeschwindigkeitsschätzung und anschließend die Generierung von Radar-Gierwinkelmessungen in Form von instantanen Verfahren entwickelt. Dies ermöglicht sowohl die Schätzung der Eigengeschwindigkeit und der Gierwinkel, nach einer Initialisierung, mit einem einzelnen Radarscan. Die Fusion beider Stützinformationen mit Inertialsensordaten sowie optionaler Barometermessungen resultiert in einem robusten vielseitig einsetzbaren lokalen Navigationssystem. Es kann eine frei konfigurierbare Anzahl an Radarsensoren eingesetzt werden wobei auch die extrinsische Kalibrierung eines jeden Sensors online geschätzt wird. So ist eine aufwendige, exakte Kalibrierung des Sensorsetups nicht notwendig und Änderungen der Kalibrierung durch mechanische Beanspruchung können kompensiert werden. Die Latenz des Radarscans wird mittels Zustandserweiterung modelliert und stellt so eine Echtzeitverarbeitung sicher. Die Formulierung

mittels eines approximierten Radarklons beeinträchtigt die Genauigkeit nicht, verbessert jedoch die Laufzeit.

Die umfangreiche Evaluation in Innenräumen und urbanen Umgebungen außerhalb von Gebäuden sowie der Vergleich zu einem etablierten Ansatz der visuellen-inertialen Odometrie zeigen die hohe Genauigkeit des entwickelten Systems. So werden durchweg bessere Ergebnisse als bei VIO, sogar in guten Sichtbedingungen erreicht. Auch in schlechten Sichtbedingungen erreicht die Radar-inertiale Navigation sehr genaue Ergebnisse. Zusätzlich ist der entwickelte Ansatz x-RIO sehr effizient und erreicht Ausführungszeiten, die um ein vielfaches schneller als Echtzeit sind. Dies wird sogar auf einem Embedded-Computer mit begrenzten Ressourcen erreicht. Die Demonstrationen autonomer Flüge in herausfordernden Umgebungen wurde in einem Bürogebäude und in dichtem Nebel mit Sichtweiten unter einem Meter erfolgreich durchgeführt und beweist die praktische Anwendbarkeit. Die Radar-inertiale Navigation stellt somit ein lokales Navigationssystem dar, sowohl in Umgebungen ohne GNSS-Empfang als auch bei schlechten Sichtbedingungen eingesetzt werden kann.

6 Multisensorielle Navigation in schwierigen Bedingungen

In diesem Kapitel wird die Navigation in schwierigen Sichtbedingungen mit Radar-basierter Multisensor-Navigation betrachtet. Hierfür wird im ersten Teil die Erweiterung von Kamera-basierten Ansätzen mit Radar-Eigengeschwindigkeitsmessungen präsentiert. Im zweiten Teil erfolgt die Erweiterung der Radar-inertialen Navigation mit GNSS zu einem globalen Navigationssystem.

6.1 Erweiterung Kamera-basierter Ansätze

Eine weitverbreitete und etablierte Möglichkeit zur Navigation unter Nutzung von Kamera und Inertialsensorik ist visuelle inertiale Odometrie (VIO). VIO wurde bereits erfolgreich zur online Navigation von UAVs eingesetzt [12, 110, 140]. Gute Sichtbedingungen sowie eine Szene mit ausreichend vielen Features sind Grundvoraussetzung für gute Ergebnisse von VIO. Abbildung 6.1 veranschaulicht den Unterschied von visuellen (obere Zeile) und thermalen (untere Zeile) Bildern in verschiedenen Bedingungen. So können Thermalkameras sogar in schlechten Sichtbedingungen durch Nebel, siehe Abbildung 6.1b, die Umgebung abbilden. Analog zu VIO, beschreibt thermale inertiale Odometrie (TIO) die Fusion von Thermalkamerabildern und Inertialsensorik. Insbesondere Innenräume wie zum Beispiel Gänge, siehe Abbildung 6.1c, weisen häufig wenig Temperaturgradienten auf, was eine robuste Bewegungsschätzung mit TIO schwierig gestaltet. Zusätzlich weisen Thermalkameras, wie auch die in dieser Arbeit eingesetzte, sogenannte Non-Uniformity-Corrections (NUC) auf. Diese regelmäßige Selbstkalibrierung dauert bis zu 0,75 s und stellt eine erhebliche Herausforderung für TIO Ansätze dar [14], weil während den NUCs keine Bilder aufgenommen werden können.



(a) Gute Bedingungen (b) Nebel (c) Dunkler Flur

Abbildung 6.1: Vergleich von visuell-thermalen-Bildpaaren in verschiedenen Sichtbedingungen. Die obere Zeile zeigt die visuellen Bilder und die untere Zeile die Thermalbilder.

Die Radar-inertiale Navigation aus Kapitel 5 stellt eine Möglichkeit zur genauen und robusten Navigation in erschwerten Sichtbedingungen dar. Hier haben sich die Hauptfehlerquellen der geschätzten Navigationslösung im driftenden Gierwinkel sowie der z-Komponente der Position gezeigt. Der Gierwinkel konnte mittels Manhattan-Welt Annahmen robust gestützt werden und die z-Komponente mittels eines Barometers. VIO oder TIO zeigen dagegen eine sehr hohe Güte bei der Schätzung des Gierwinkels und der z-Komponenten und weisen in der Regel Skalenfehler in Bewegungsrichtung auf. Dies zeigt sich vor allem bei monokularen Ansätzen.

Daher untersucht dieser Abschnitt die Erweiterung von VIO und TIO mit Radar-Eigengeschwindigkeitsmessungen. Radar-Eigengeschwindigkeit weist keine Skalenfehler auf und ist insbesondere in Bewegungsrichtung sehr genau. Dadurch können Skalenfehler von monokularen Setups erheblich verringert werden. Es werden jedoch ausreichend Objekte innerhalb des Radarscans benötigt. Radar-visuelle-inertiale Odometrie (RVIO) funktioniert auch in Szenen mit wenigen Features oder schwierigen Sichtbedingungen. Radar-thermale-inertiale Odometrie (RTIO) ermöglicht robuste Zustandsschätzung auch in Umgebungen mit geringen Temperaturgradienten. Beide Ansätze profitieren

von den Eigenschaften von VIO oder TIO in jeweils vorteilhaften Bedingungen und können so Bereiche ohne Objekte innerhalb des Radarscans überbrücken. Es werden somit die komplementären Eigenschaften beider Ansätze ausgenutzt.

Im Rahmen dieser Arbeit wurde die etablierte Pipeline ROVIO [12] um die Fusion mit Radar-Eigengeschwindigkeit erweitert. Die vorgeschlagene Pipeline robuste Radar visuelle-inertiale Odometrie (RRxIO) wurde in Doer et al. [180] veröffentlicht.

Im Folgenden werden zunächst verwandte Arbeiten analysiert, anschließend wird die erweiterte VIO-Pipeline und die notwendigen Erweiterungen näher vorgestellt. Darauf folgt eine ausführliche Evaluation zunächst mit Simulation und anschließend mit selbst erstellten Datensätze verschiedenen Kombinationen aus den visuell-thermal-Radar-inertialen-Domänen. Die folgenden Ausführungen basieren auf der Veröffentlichung Doer et al. [180].

6.1.1 Verwandte Arbeiten

Visuelle-inertiale Odometrie

Visuelle-inertiale Odometrie (VIO) wurde im vergangenen Jahrzehnt umfangreich erforscht. Dabei haben sich zwei Hauptgruppen herausgebildet: Filter-basierte Ansätze [12, 46, 81, 98, 140] und Optimierungs-basierte Ansätze [43, 80, 110]. Eine umfangreiche Analyse von sechs Open-Source Ansätzen zur Navigation von fliegenden Systemen wurde in [33] durchgeführt. Hierin folgerten Delmerico et. al., dass VINS [110] die höchste Genauigkeit und ROVIO [12] den besten Tradeoff zwischen Genauigkeit und Laufzeit erzielt. Da diese Arbeit auf Navigationsansätze abzielt, die auch auf Embedded-Computern mit begrenzter Rechenleistung ausgeführt werden können, wird im Folgenden ROVIO als Grundlage für das vorgeschlagene System verwendet [180]. VINS wird dagegen zur Evaluation, als Vergleich zu einem sehr genauen VIO-Ansatz, eingesetzt. Der Einsatz von VIO eignet sich nur bei guten Sichtbedingungen. Erschwerte Bedingungen beispielsweise durch Dunkelheit oder Nebel lassen sich mit reinen VIO-Ansätzen nicht lösen. Die Kombination mit Radar-Eigengeschwindigkeitsmessungen, stellt jedoch eine vielversprechende Möglichkeit dar, um auch mit solchen Bedingungen umzugehen.

Thermale-inertiale Odometrie

Analog zu VIO, wurde in den letzten Jahren thermale-inertiale Odometrie (TIO) als Lösung auch für schwierige Sichtbedingungen untersucht [32, 69, 72, 126, 160]. Dabei hat sich die Fusion mit Inertialsensorik als sehr gewinnbringend herausgestellt, da Thermalbilder im Vergleich zu visuellen Bildern insbesondere in Innenräumen deutlich weniger robuste Features aufweisen, siehe auch Abbildung 3.1. Dies wird durch die Ansätze [69, 72, 160] kompensiert, indem die Bildintensitäten in direkter Nachbarschaft zum detektierten Feature genutzt werden. Um eine höhere Dynamik abzubilden, nutzen alle drei Ansätze [69, 72, 160] nicht die in der Bildverarbeitung üblichen 8 bit Graustufenwerte, sondern die rohen radiometrischen Messungen der Thermalkamera in Form von 14 bit. Somit wird eine deutlich höhere Dynamik erzielt da anstatt von einem Wertebereich 0 – 255 (8 bit) ein Bereich 0 – 16384 (14 bit) genutzt werden kann. Khattak et al. [69] haben daher die Erweiterung von ROVIO zur Verarbeitung von 14 bit Graustufenwerten vorgeschlagen. Um Rechenzeit zu sparen und einen allgemeinen Ansatz sowohl für die visuelle als auch die thermale Domäne bereitzustellen wird im Rahmen dieser Arbeit der Einsatz von 8 bit Daten umgesetzt [180]. Auch die Anwendung von Ansätzen des maschinellen Lernens wurden vorgeschlagen [126, 160]. Während Saputra et al. [126] ein neuronales Netz einsetzen, um Thermalbilder für die weitere Verarbeitung zu optimieren, verfolgen Zhao et al. [160] den Ansatz mit Hilfe von Deep Learning einen robusten Feature Detektor speziell für Thermalbilder zu entwickeln. Da jedoch beide Ansätze rechenintensive neuronale Netzwerke nutzen, ist eine Echtzeitfähigkeit auf einem Embedded-Computer mit limitierter Rechenleistung nicht möglich, was jedoch das Ziel dieser Arbeit ist.

Radar-inertiale Odometrie

Wie bereits in Unterabschnitt 5.1 analysiert, stellt die Radar-inertiale Navigation einen Ansatz dar, der das Szenario der schlechten Sichtbedingungen löst. Ohne eine Gierwinkelstützung auf Basis von Manhattan-Welt Annahmen tritt jedoch bei dem Verfahren aus Kapitel 5 auf Grund des Gierratenoffsetdrifts ein wachsender Fehler des Gierwinkels auf. Da der Gierratenoffset jedoch durch

VIO oder TIO sehr gut geschätzt wird, ist die Fusion mit VIO oder TIO, also ein multi-Domänen Ansatz, sehr vielversprechend.

Multisensorielle Odometrie

Weitere Multiomänen Ansätze in dem Kontext wurden in [71, 82, 97, 135] betrachtet. Diese erzielen durch die Verwendung mehrerer Domänen als nur VIO oder TIO bessere Ergebnisse. So wurden in [135] die Fusion von einer Kamera und Lidar vorgeschlagen. Jedoch kann das Szenario mit Nebel oder Rauch durch diesen Ansatz nicht gelöst werden. Khattak et al. [71] verfolgten dabei den Ansatz, Features sowohl aus visuellen als auch Thermalbildern gleichzeitig zu nutzen. Somit ist ein Einsatz auch in schwierigen Bedingungen möglich. Im Rauch oder Nebel erfolgt dann jedoch die Navigation ausschließlich mit Hilfe der Thermalbilder was insbesondere in Innenräumen, wie oben beschrieben, schwierig sein kann. Der Einsatz von Radar wurde bisher nur in [82, 97] betrachtet. Mostafa et al. [97] kombinierten dabei jedoch nur eindimensionale Radargeschwindigkeitsmessungen mit visuellen optischen Fluss Messungen, Radarhöhenstützung und GPS. Hierbei stellt jedoch die eindimensionalen Radargeschwindigkeitsmessungen nur eine geringe Stützinformation bereit was dein Einsatz in schwierigen Sichtbedingungen nicht löst. Liang et al. [82] nutzen dagegen zweidimensionale Radargeschwindigkeitsmessungen und fusionieren diese Kalman-Filter-basiert mit GNSS-, Lidar-, Stereokamera- und Inertialsensordaten. Wie auch schon im Ansatz von [97], stellt Radar bei [82] nur eine weitere Stützinformation dar, die nur horizontal stützt. Da hier jedoch die Navigation von UAS im dreidimensionalen Raum betrachtet wird, ist eine horizontale Stützung nicht ausreichend.

Fazit

Es zeigt sich also eine Lücke im Stand der Technik bezüglich der Fusion von Radar mit VIO (RVIO) und TIO (RTIO) was durch diese Arbeit geschlossen wird [180]. So wird ROVIO [12] erweitert, um 3D Radar-Eigengeschwindigkeitsmessungen zu fusionieren. Der entstehende Ansatz wird robuste Radar visuelle-inertiale Odometrie (RRxIO) genannt [180].

6.1.2 Robuste visuelle-inertial Odometrie (ROVIO)

Robuste visuelle-inertiale Odometrie (ROVIO) ist ein Ansatz, der auf einem iterativen Kalman-Filter basiert [12]. Im Bild werden zunächst FAST [120] Features detektiert. Um die Positionen der Features herum werden dann die Bildintensitäten in Form von Patches extrahiert. Dies sind die direkten Graustufenwerte, die in einem quadratischen Bereich, zum Beispiel 3×3 pixel, um die Feature-Position aus dem Bild extrahiert werden. ROVIO nutzt daher einen sogenannten direkten visuellen Odometrie-Ansatz, der den photometrischen Fehler minimiert. Dies stellt einen robusteren Ansatz als klassische Ansätze dar, die den Rückprojektionsfehler minimieren, weil so alle Informationen in direkter Umgebung des Features genutzt werden [12]. Für mehr Robustheit erfolgt die Extraktion der Patches auf einer Bildpyramide, wobei jede Stufe die Auflösung des Bildes halbiert. So können zum Beispiel auch Kanten im Bild robust getrackt werden. Vor der eigentlichen Verarbeitung als Innovationsterm im Kalman-Filter-Update, werden die Bild-Patches mit Hilfe des aktuellen Filterzustands affin transformiert [12]. So werden Verzerrungen durch Rotationen kompensiert. Im Filter-Update wird dann das Residuum auf Basis des prädizierten Patches und der tatsächlichen Patches im Bild aufgestellt [12]. Beleuchtungswechsel werden mit einem affinen Intensitätsmodell kompensiert [12].

ROVIO nutzt eine sogenannte robozentrische Formulierung des Schätzproblems, was bessere Eigenschaften der Beobachtbarkeit zur Folge hat [12]. Folglich werden alle Zustände im körperfesten Koordinatensystem geschätzt. Der Zustandsvektor $\boldsymbol{x}$ zum Zeitpunkt k ist dabei gegen durch [12]

$$\boldsymbol{x}_k = \begin{bmatrix} \boldsymbol{p}^b & \boldsymbol{v}^b & \boldsymbol{q}_o^b & \boldsymbol{b}_a & \boldsymbol{b}_g & \boldsymbol{l}_{bc}^b & \boldsymbol{q}_c^b & \boldsymbol{x}_f \end{bmatrix} \tag{6.1}$$

mit der Position $\boldsymbol{p}^b$, der Geschwindigkeit $\boldsymbol{v}^b$, der Lage $\boldsymbol{q}_o^b$, den Offets des Beschleunigungs- $\boldsymbol{b}_a$ und des Drehratensensors $\boldsymbol{b}_g$ sowie der extrinsischen Kalibrierung der Kamera bestehend aus der Translation $\boldsymbol{l}_{bc}^b$ und Rotation $\boldsymbol{q}_c^b$ zwischen dem Kamera und dem köperfesten Koordinatensystem. Der Featurevektor $\boldsymbol{x}_f$ für den Fall von N-Features besteht aus [12]:

$$\boldsymbol{x}_f = \begin{bmatrix} \boldsymbol{\mu}_0^c & \dots & \boldsymbol{\mu}_N^c & \rho_0 & \dots & \rho_N, \end{bmatrix}. \tag{6.2}$$

und parametrisiert die geschätzte Feature-Position mit dem Richtungsvektor $\boldsymbol{\mu}$ und inverser Tiefen ρ relativ zum körperfesten Koordinatensystem. Die Verarbeitung des Filters erfolgt bildgetrieben [12]. Das heißt, Propagation erfolgt erst, sobald ein neues Kamerabild vorliegt. Dann wird der Zustand bis zum Zeitstempel des Bildes propagiert und das Filterupdate ausgeführt. Somit ist die Update-Rate des Filterzustands im Gegensatz zur Radar-inertialen Navigation aus Abschnitt 5, nur so hoch wie die Bildrate, also typischerweise 20 Hz anstelle von 200 Hz.

6.1.3 Robuste Radar visuelle-inertiale Odometrie (RRxIO)

Abbildung 6.2 zeigt das Blockdiagramm von der entwickelten Pipeline RR-xIO basierend auf Doer et al. [180]. Es erfolgt zunächst die Schätzung der Radar-Eigengeschwindigkeit $\boldsymbol{v}^r$ mit Unsicherheit $\mathbf{P}_r$ mit Hilfe des Ansatzes aus Abschnitt 5.2. Anschließend wird diese als weiteres Messupdate im EKF von ROVIO mit den Daten der IMU und der Kamera fusioniert.

Für das Geschwindigkeitsupdate muss dabei die Radar-Eigengeschwindigkeit in das körperfeste Koordinatensystem transformiert werden. Dabei wird angenommen, dass die extrinsische Kalibrierung des Radarsensors $(\boldsymbol{l}^b_{br}, \boldsymbol{q}^b_r)$ bekannt ist. Die Umformung von (5.30) ergibt [180]

$$\boldsymbol{v}^b_r = \mathbf{C}^b_r \boldsymbol{v}^r - \left[\omega^b_m - \boldsymbol{b}_g\right] \times \boldsymbol{l}^b_{br} \tag{6.3}$$

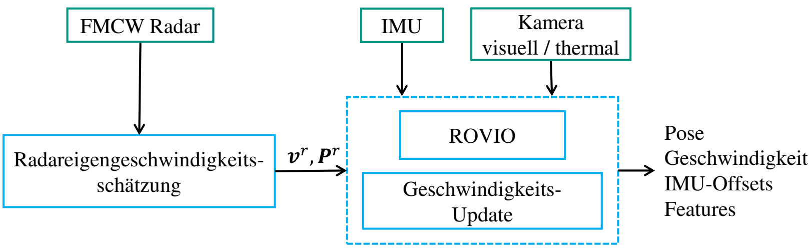


Abbildung 6.2: Blockdiagramm der entwickelten Pipeline: Robuste Radar visuelle/thermal-inertiale Odometrie (RRxIO).

mit der gemessenen Drehrate der IMU $\boldsymbol{\omega}_m^b$ und dem geschätzten Offset der Drehrate $\boldsymbol{b}_g$ aus dem Filterzustand. Da hier MEMS IMUs eingesetzt werden, wird die Drehrate der Erde vernachlässigt. Anschließend ergibt sich die Messgleichung der Geschwindigkeit im körperfesten Koordinatensystem $\boldsymbol{v}_m^b$ zu

$$\boldsymbol{v}_m^b = \boldsymbol{v}_r^b + \boldsymbol{v}_{v_r}, \tag{6.4}$$

wobei das Messrauschen mit Gaußschen Rauschen $\boldsymbol{v}_{v_r}$ modelliert und durch die geschätzte Kovarianz $\mathbf{P}_r$ parametrisiert wird [180]. Linearisierung ergibt die Jacobi-Matrix für N-Features im Filterzustand:

$$\mathbf{H}_r = \begin{bmatrix} \mathbf{0}_{3\times3} & \mathbf{I}_3 & \mathbf{0}_{3\times6} & -\left[\boldsymbol{l}_{br}^b\right]\times & \mathbf{0}_{3\times4N} \end{bmatrix}. \tag{6.5}$$

Wie bereits in Unterabschnitt 5.4.3 diskutiert ist der Einfluss des Drehratenoffsets auf das Geschwindigkeitsresiduum sehr gering. Daher wird hier die Jacobi-Matrix approximiert durch [180]:

$$\mathbf{H}_r \approx \begin{bmatrix} \mathbf{0}_{3\times3} & \mathbf{I}_3 & \mathbf{0}_{3\times6} & \mathbf{0}_{3\times(4N+3)} \end{bmatrix}. \tag{6.6}$$

Ein Mahalanobis-Distanz-Test, siehe Abschnitt A.1, stellt sicher, dass Outlier herausgefiltert werden.

6.1.4 Evaluation mit EuRoC-Datensätzen

Zunächst wird die vorgeschlagene Pipeline RRxIO mit einem der bekanntesten UAV-Flugdatensätzen evaluiert. Der *EuRoC-MAV-Datensatz* [17] besteht aus elf UAV-Flügen mit visuell-inertialen Sensordaten sowie jeweils einer Referenz zur Evaluation. Im Folgenden werden die Daten der IMU sowie der linken Kamera genutzt. Da dieser Datensatz keine Radarsensordaten enthält, werden Radar-Eigengeschwindigkeitsmessungen simuliert. So kann der Einfluss hiervon auf die Güte der Navigation mit realen visuell-inertialen Daten analysiert werden. Für die Simulation der Radar-Eigengeschwindigkeit wird Gaußsches Rauschen mit einer Standardabweichung von $[0.05, 0.075, 0.12]$ m/sec simuliert. Dieses wurde zuvor in Datensätzen mit Motion-Capture Referenzsystem und dem realen FMCW-Radar bestimmt, wie in Abschnitt 5.2.5 präsentiert ist. Wie auch im realen Sensor-Setup beträgt die Messrate 10 Hz.

Die Evaluation erfolgt nach dem gleichen Verfahren wie bei der Radar-inertialen Navigation in Abschnitt 5.4.5. So werden die geschätzten Trajektorien zur Referenz mit *pos-yaw* ausgerichtet [145] und anschließend der absolute Translationsfehler mittels [159] berechnet. Zusätzlich wird auch der Echtzeit-Faktor des *Intel-NUC-i7* analysiert, also die Datensatzlänge geteilt durch die Laufzeit des Algorithmus'.

Tabelle 6.1 zeigt die Ergebnisse für die *EuRoC-MAV-Datensätze Machine-Hall* und *Vicon-Room*. Dabei wurden folgende Ansätze evaluiert:

- VINS [110]: Nach [33] eine der genausten Open-Source VIO Pipelines.
- ROVIO [12]: Nach [33] die Pipeline mit dem besten Tradeoff zwischen Genauigkeit und Laufzeit. Es werden 25 Features im Filterzustand verwendet.
- RRxIO [180]: Die Weiterentwicklung von ROVIO mit der Fusion von Radar-Eigengeschwindigkeitsmessungen. Hierbei wird die Anzahl an Features N des Filterzustands, siehe (6.1), variiert.

	VINS	ROVIO	RRxIO (vorgeschlagen)		
Dataset		N=25	N=25	N=15	N=10
Machine-Hall-01	0, 25	0, 46	**0, 21**	0, 27	0, 33
Machine-Hall-02	**0, 10**	0, 24	0, 23	0, 20	0, 20
Machine-Hall-03	0, 20	0, 37	0, 20	0, 20	**0, 19**
Machine-Hall-04	0, 30	0, 75	**0, 21**	0, 51	0, 79
Machine-Hall-05	0, 27	0, 70	**0, 23**	0, 68	0, 74
Vicon-Room1-01	0, 10	**0, 08**	**0, 08**	0, 13	0, 13
Vicon-Room1-02	0, 11	**0, 09**	0, 10	0, 17	0, 17
Vicon-Room1-03	0, 16	0, 14	0, 12	**0, 11**	0, 18
Mittlerer ATE (m)	0, 19	0, 36	0, 17	0, 28	0, 34
Echtzeitfaktor Faktor	1.1	7.4	7.1	13.2	19.3

Tabelle 6.1: Ergebnisse *EuRoC-MAV-Datensatz* [17]: Alle Ergebnisse wurden zur Referenz ausgerichtet und hinsichtlich des absoluten Translations-RMS-Fehlers (ATE) evaluiert, nach Doer et al. [180]. Das beste Ergebnis pro Datensatz ist hervorgehoben. N beschreibt dabei die Anzahl an Features im Filterzustand.

RRxIO verbessert die mittlere Positionsgenauigkeit in allen Konfiguration im Gegensatz zu ROVIO. Dies zeigt sich im mittleren Positionsfehler und insbesondere in den Datensätzen der *Machine-Hall*. Diese weisen im Gegensatz zu den *Vicon-Room*-Datensätzen eine höhere Szenetiefe auf. Dies führt zu höheren Fehlern als bei Szenen mit geringer Tiefe wie bei den Datensätzen *Vicon-Room1*. RRxIO mit $N = 25$ Features erreicht sogar einen besseren mittleren Fehler als VINS (0,17 m vs. 0,19 m) bei einer knapp siebenfach schnelleren Laufzeit ($7,1$ vs. $1,1$). RRxIO mit $N = 10$ Feature erreicht eine ähnliche Genauigkeit wie ROVIO mit $N = 25$ Features wobei die Laufzeit um mehr als den Faktor zwei verbessert wird.

Daher ergeben sich die folgenden zwei Konfigurationsmöglichkeiten, je nach Ziel der Anwendung. So kann RRxIO mit der gleichen Anzahl an Features verwendet werden und erzielt dann deutlich bessere Ergebnisse als ROVIO, bei ähnlicher Verarbeitungszeit. Oder RRxIO wird mit weniger Features betrieben, erzielt dann ähnliche Genauigkeiten wie ROVIO (mit einer höheren Anzahl an Features), bei stark verbesserter Laufzeit.

6.1.5 Evaluation mit Realläufen

Im Folgenden wird RRxIO ausführlich mit Realläufen evaluiert und mit verschiedenen weiteren Ansätzen auf umfangreichen Datensätzen verglichen. Dabei werden die Kombinationen Radar-visuell-inertial und Radar-thermal-inertial von RRxIO gegen Radar-inertial, visuell-inertial und thermal-inertial gegeneinander evaluiert. So können die Stärken und Schwächen der Ansätze und Domänen verglichen werden.

Hardware-Aufbau

Für die Realläufe wurde eine leicht modifizierte Version der Multisensorplattform aus Unterabschnitt 3.1 verwendet. Diese ist in Abbildung 6.3 auf der linken Seite dargestellt. Im Gegensatz zur Multisensorplattform wurde hier nur ein nach vorne gerichteter Radarsensor (TI IWR6843AOP) sowie eine Mono-Graustufenkamera eingesetzt. Zusätzlich ist die Plattform auch mit Markern für das Motion-Capture System ausgestattet. Für die geflogenen Datensätze wurde die Flugplattform, siehe Abbildung 6.3 rechts, eingesetzt. Mit

Multisensorplattform (getragen Datensätze) | **Flugsystem mit Multisensorplattform (geflogenen Datensätze)**

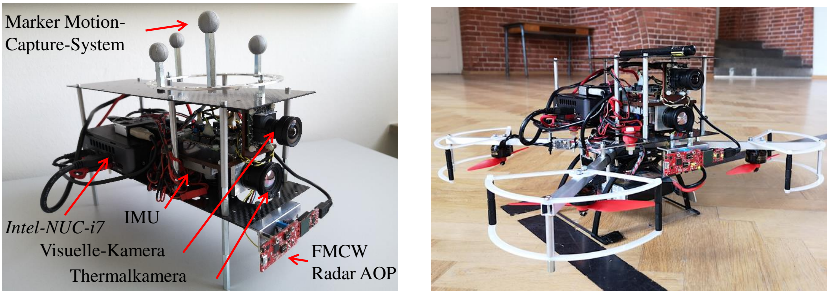


Abbildung 6.3: Modifizierter Hardware-Aufbau für getragene Datensätze (links) und geflogene Datensätze (rechts) nach Doer et al. [180].

Hilfe des Hardwareaufbaus wurden umfangreich Datensätze in verschiedenen Umgebungen und Sichtbedingungen aufgezeichnet, die im nächsten Abschnitt zusammengefasst werden.

Überblick Datensätze

Abbildung 6.4 zeigt eine Übersicht der zur Evaluation eingesetzten Datensätze jeweils mit einem visuell-thermalen Bildpaar. Dabei sind auf der linken Seite und Mitte die Datensätze mit Motion-Capture und Pseudo-Referenz sowie auf der rechten Seite Datensätze in schlechten Sichtbedingungen gezeigt. Die Motion-Capture-Referenz Datensätze wurden im Motion-Capture Lab des IRS aufgezeichnet sowohl in guten als auch in schlechten Sichtbedingungen, die durch Dunkelheit erzeugt wurden. Die Datensätze mit Pseudoreferenz umfassen verschiedene Umgebungen in Innenräumen und außerhalb von Gebäuden. Die Datensätze in schlechten Sichtbedingungen wurden bei Nacht oder unter Einsatz einer Nebelmaschine aufgezeichnet und decken Innenräume und Außenbereiche ab. Im Folgenden werden zunächst die Ergebnisse mit Referenz und dem gleichen Evaluationsschema wie im Abschnitt 6.1.4 dargestellt. Anschließend folgt eine Evaluation in schlechten Sichtbedingungen. Hierbei wird nur der finale Positionsfehler evaluiert, da keine Referenz mit Stand der Technik Methoden erstellt werden konnte.

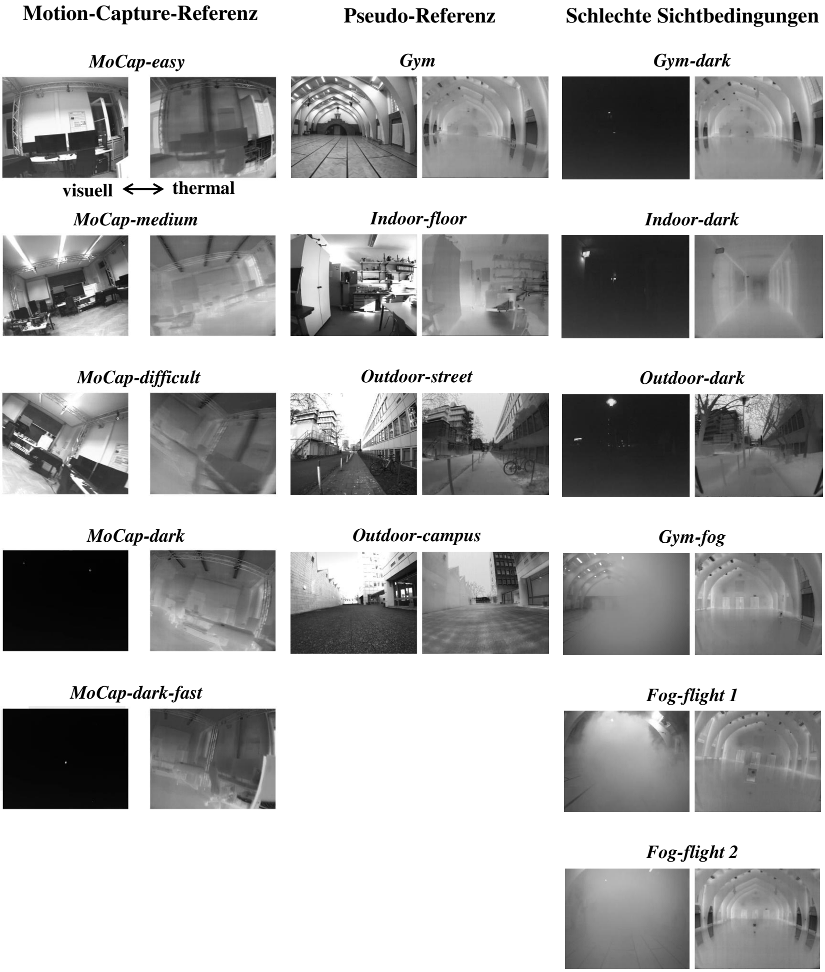


Abbildung 6.4: Überblick der Datensätze: Motion-Capture-Referenz (linke Spalte), Pseudo-Referenz (Mitte) und schlechte Sichtbedingungen (rechte Spalte) jeweils mit einem visuell-thermalen Bildpaar.

Evaluation mit Referenz: Methodik

Die Datensätze mit Referenz wurden mit Ausnahme der *Mocap-Dark-Datensätze* bei guten Sichtbedingungen aufgezeichnet, um eine breite Evaluation durchzuführen. Es handelt sich hierbei um neun Datensätze die Umgebungen wie das Motion-Capture Lab, Innenräume und urbane Gebiete abdecken.

Die Referenz wurde dabei bei den fünf *MoCap-Datensätzen* mit Hilfe des Motion-Capture Systems sowie der Inertialsensordaten mittels einer Batch-Optimierung mit der Toolbox *vicon2gt* [47] erzeugt. Die *Pseudo-Referenz-Datensätze*, dagegen, wurden mit einem visuell-inertialen SLAM erstellt. Dabei wurde VINS [110] mit aktivierten Schleifenschlüssen eingesetzt. Bei den Datensätzen wurde auf regelmäßige Schleifenschlüsse geachtet, damit mit dem visuell-inertialen SLAM akkurate Referenz-Trajektorien erzeugt werden können. Da es sich jedoch um ein monokulares Setup handelt, treten Skalenfehler der geschätzten Trajektorie auf. Diese wurden mit Hilfe von Referenz-Messpunkten manuell bestimmt und korrigiert.

Mit Hilfe der Referenz, können die geschätzten Trajektorien als Ganzes evaluiert werden, also der Fehler zu jedem Zeitschritt bestimmt werden, wie auch schon zur Evaluation der Radar-inertialen Navigation in Unterabschnitt 5.4.5 eingesetzt. Es werden die folgenden Ansätze und Konfigurationen evaluiert:

- X-RIO [182] (aus Kapitel 5): Evalution Radar-inertial in der Konfiguration mit nur einem Radarsensor sowie ohne Stützung durch Barometer- und Gierwinkelupdate.
- VINS [110] (ohne Schleifenschlüsse): Evaluation visuell-inertial und thermal-inertial, jeweils mit getunten Parametern.
- ROVIO [12]: Evaluation visuell-inertial und thermal-inertial mit $N = 25$ Features, jeweils mit getunten Parametern.
- RRxIO [180]: Evaluation Radar-visuell-inertial und Radar-thermal-inertial jeweils mit $N \in [25, 15, 10]$ Features.

Die Ergebnisse sind im oberen Abschnitt von Tabelle 6.2 ausführlich dargestellt und werden im Folgenden diskutiert.

	Radar-inertial	**visuell-inertial**		**Radar-visuell-inertial**			**thermal-inertial**		**Radar-thermal-inertial**		
	x-RIO	VINS	ROVIO	RRxIO (vorgeschlagen)			VINS	ROVIO	RRxIO (vorgeschlagen)		
			N=25	N=25	N=15	N=10		N=25	N=25	N=15	N=10
Motion-Capture Referenz											
MoCap-easy: 36 m, 94 s	0, 12	0, 09	0, 09	**0, 08**	0, 11	0, 10	0, 20	0, 12	**0, 08**	0, 14	0, 25
MoCap-medium: 83 m, 87 s	0, 28	**0, 08**	0, 10	0, 10	0, 08	0, 12	-	**0, 09**	0, 12	0, 13	0, 16
MoCap-difficult: 70 m, 86 s	0, 32	0, 14	0, 09	0, 09	**0, 07**	0, 21	-	0, 15	0, 16	**0, 15**	0, 20
MoCap-dark: 111 m, 135 s	0, 22	-	-	**0, 14**	0, 14	0, 14	-	0, 29	**0, 17**	0, 17	0, 24
MoCap-dark fast: 75 m, 86 s	0, 28	-	-	**0, 23**	0, 23	0, 23	0, 20	0, 20	**0, 15**	0, 16	0, 16
Pseudo-Referenz											
Gym: 74 m, 84 s	0, 12	0, 86	0, 48	**0, 11**	0, 13	0, 16	0, 51	0, 82	**0, 20**	0, 26	0, 29
Indoor-floor: 240 m, 206 s	0, 93	0, 39	**0, 29**	0, 30	0, 39	0, 48	1, 26	1, 40	0, 48	**0, 44**	0, 61
Outdoor-campus: 102 m, 100 s	0, 47	0, 62	2, 80	**0, 47**	0, 53	0, 66	1, 10	3, 94	**0, 26**	0, 42	0, 44
Outdoor-street: 262 m, 186 s	2, 11	0, 57	3, 46	**0, 57**	0, 93	1, 25	2, 27	8, 60	0, 73	**0, 64**	1, 10
Mittlerer ATE [m]	0, 54	0, 39	1, 04	**0, 23**	0, 29	0, 37	0, 94	1, 72	**0, 26**	0, 28	0, 38
Schlechte Sichtbedingungen											
Gym-fog: 74 m, 111 s	0, 61	1, 21	1, 61	**0, 32**	0, 45	0, 73	-	1, 04	**0, 39**	0, 68	0, 50
Gym-dark: 74 m, 90 s	0, 58	-	-	**0, 39**	0, 82	0, 41	-	1, 26	0, 47	0, 56	**0, 29**
Indoor-dark: 120 m, 199 s	1, 62	-	-	1, 45	1, 45	**1, 32**	-	-	0, 60	**0, 48**	1, 28
Outdoor-dark: 222 m, 173 s	1, 21	-	-	0, 31	**0, 30**	0, 32	1, 59	**0, 72**	0, 94	0, 82	0, 85
Fog-flight 1: 29 m, 72 s	0, 29	1, 72	1, 18	**0, 60**	0, 79	0, 92	-	1, 98	**0, 39**	1, 17	2, 18
Fog-flight 2: 40 m, 101 s	0, 28	-	1, 12	**1, 00**	1, 42	1, 90	-	**0, 86**	1, 24	1, 02	0, 87
Mittlerer finaler relativer Fehler [%]	0, 77	-	-	**0, 68**	0, 87	0, 95	-	1, 17	**0, 67**	0, 79	1, 00
Mittlerer Echtzeitfaktor *Intel-NUC-i7*	140	1	6	9	10, 9	13	1	7	7	10	12
Mittlerer Echtzeitfaktor *Up-Core*	21	-	0, 6	0, 7	1	1, 2	-	0, 6	0, 6	0, 9	1, 2

Tabelle 6.2: Evaluation mit Referenz (oben und Mitte) hinsichtlich des mittleren Translationsfehlers sowie in schlechten Sichtbedingungen (unten) bezüglich des relativen finalen Positionsfehlers basierend auf Doer et al. [180]. Da bei schlechten Sichtbedingungen keine Referenz mit VINS erstellt werden konnte, wird hier der finale Positionsfehler evaluiert. Das beste Ergebnis pro Datensatz ist für die visuelle Domäne in blau und für die thermale Domäne in grün markiert. Ergebnisse, die divergiert sind oder sehr große Fehler aufweise, sind mit einem Minus gekennzeichnet.

Evaluation mit Referenz: Visuelle Kamera

In der visuellen Domäne zeigen sich ähnliche Ergebnisse aller Ansätze bei den ersten drei *MoCap*-Datensätzen. Analog zu den *Vicon-Room-Datensätzen* aus Abschnitt 6.1.4, liegen hier geringe Szenentiefen und gute Sichtbedingungen vor, sodass VINS und ROVIO sehr akkurate Ergebnisse liefern. Eine weitere Verbesserung durch Radar-Eigengeschwindigkeit ist bei den bereits erzielten Genauigkeiten nicht möglich. *MoCap-Dark* und *MoCap-Dark-Fast* dagegen, konnten nur mit RRxIO erfolgreich prozessiert werden. Da hier das Licht ausgeschaltet war, gab es nicht mehr genügend visuelle Features. Die Fusion mit Radar-Eigengeschwindigkeit ist dabei sehr vorteilhaft. So können beide Datensätze mit RRxIO in der Radar-visuellen-inertialen-Domäne erfolgreich verarbeitet werden. Auf Grund der geringen Anzahl an visuellen Features ergeben sich dabei jedoch die gleichen Ergebnisse für alle drei Konfigurationen von RRxIO. Die vier Datensätze mit Pseudo-Referenz sind deutlich herausfordernder. Insbesondere bei den Datensätzen mit großer Szenetiefe (*Gym*, *Outdoor-Campus* und *Outdoor-Street*) erzielt RRxIO deutlich bessere Ergebnisse als VINS und ROVIO. Insgesamt erreicht RRxIO mit $N = 25$ Features im Schnitt um den Faktor vier geringere Positionsfehler als ROVIO mit der gleichen Anzahl an Features. Selbst RRxIO mit $N = 10$ Features erzielt um mehr als den Faktor zwei geringere Positionsfehler als ROVIO mit $N = 25$ Features. Alle drei RRxIO-Konfigurationen erzielen im Schnitt bessere Positionsgenauigkeiten als VINS und bei $N = 25$ ist RRxIO fast doppelt so genau wie VINS. RRxIO erreicht mit $N = 25$ Features einen Positionsfehler, der um 37 % geringer ist ist als mit $N = 10$ Features.

Evaluation mit Referenz: Thermalkamera

In der thermalen Domäne zeigt sich der Vorteil von RRxIO gegenüber den rein thermal-inertialen Ansätzen sehr deutlich. So können drei der neun Datensätze von VINS nicht prozessiert werden. ROVIO ist als direktes VIO-Verfahren gerade bei Thermalbildern gegenüber indirekten Verfahren, wie zum Beispiel VINS, überlegen. So kann ROVIO alle neun Datensätze erfolgreich verarbeiten. Auch bei Thermalbildern stellt sich die Fusion mit Radar-Eigengeschwindigkeit als sehr vorteilhaft heraus. RRxIO erzielt in der thermalen Domäne mit $N = 25$ Features eine sechsfach bessere Genauigkeit als ROVIO sowie vierfach

bessere Ergebnisse als VINS. Auch in der Konfiguration mit $N = 10$ Features erreicht RRxIO in der thermalen Domäne mehr als vierfach bessere Genauigkeit als ROVIO und knapp dreifach bessere Genauigkeiten als VINS. RRxIO erreicht mit $N = 25$ Features einen Positionsfehler, der um 32 % geringer ist als mit $N = 10$ Features. Im Vergleich zu RRxIO der visuellen Domäne ergeben sich in der thermalen Domäne im Schnitt ähnliche Genauigkeiten über alle drei Konfigurationen.

Evaluation mit Referenz: Radar

Die Radar-inertiale Navigation (x-RIO) wurde hier mit einem einzelnen Radarsensor und ohne die Stützung durch Barometer- und Gierwinkelmessungen evaluiert und somit nur die reine Radar-inertialen Domäne betrachtet. Es zeigt sich, dass x-RIO im Schnitt bessere Genauigkeiten als ROVIO in der visuell-inertialen und thermal-inertialen Kombination erzielt. RRxIO erreicht jedoch mit allen Konfigurationen sowie mit visuellen als auch thermalen Bildern bessere Genauigkeiten. RRxIO mit $N = 25$ Features erzielt eine um den Faktor zwei höhere Genauigkeit als x-RIO. Die Fusion mit Kameradaten stellt somit einen wertvollen Informationsgewinn dar, der zu einer Verbesserung der Lokalisierungsgenauigkeit führt.

Evaluation mit Referenz: Relative Fehler

Abbildung 6.5 zeigt die Analyse der relativen Fehler für den Datensatz *Gym*. Dieser wurde nach der Fehlermetrik von Geiger et al. [45] berechnet und evaluiert Trajektorienabschnitte mit den Intervallen $[12, 24, 36, 48, 60]$m. Das heißt, über den gesamten Datensatz werden die Fehler im Abstand dieser Intervalle jeweils berechnet. Dabei wird der jeweilige Trajektorienabschnitt mit Hilfe des ersten Samples zur Referenz ausgerichtet und anschließend der Fehler berechnet. So können Fehler der Position und des Gierwinkels sehr gut getrennt voneinander betrachtet werden, da ein einmaliger Gierwinkelfehler nicht die gesamte Folgetrajektorie beeinflusst. Der waagerechte Strich im Innern der rechteckigen Boxen, auch Whiskers genannt, markiert dabei den Median und die obere und untere Kante jeweils das obere und untere Quantil der Fehler und deckt somit 50 % der Fehler ab. Es wird ein Vergleich von ROVIO mit $N = 25$ Features (visuell: blau, thermal: rot) und RRxIO mit $N = 15$ (visuell:

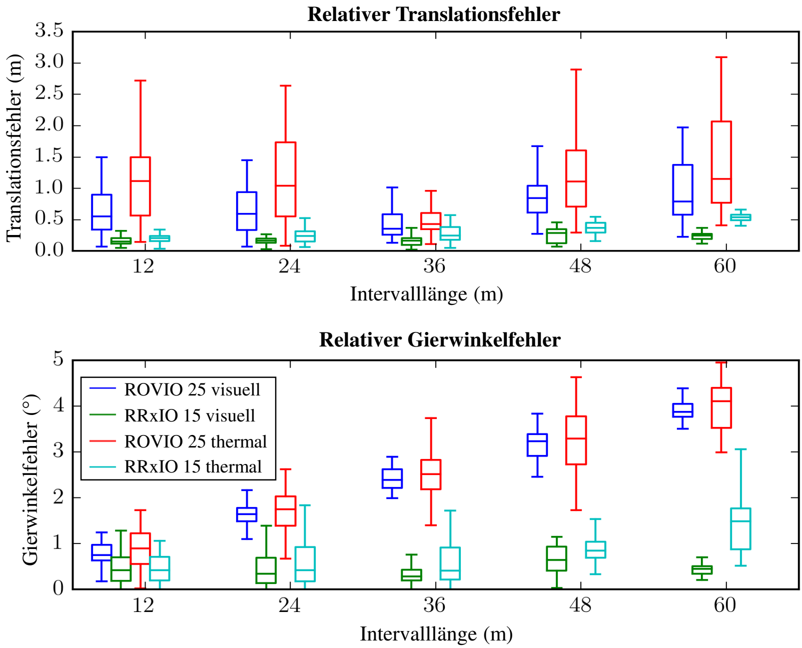


Abbildung 6.5: Boxplot des relative Fehlers des *Gym*-Datensatzes der Translation (oben) und des Gierwinkels (unten) nach Doer et al. [180]. Die rechteckigen Boxen markieren dabei das ober und untere Quantil, schließen also 50 % der Fehler ein. Der waagerechte Strich im Innern der Boxen ist der Median des Fehlers.

grün, thermal: blau) durchgeführt. Dabei zeigt sich die Verbesserung durch die Fusion mit Radar-Eigengeschwindigkeit sehr deutlich sowohl im Translations- als auch im Gierwinkelfehler. Beide Fehler wachsen zwar in Abhängigkeit der zurückgelegten Strecke, fallen aber bei RRxIO im Vergleich zu ROVIO deutlich geringer aus. RRxIO erzielt dabei für die visuelle Domäne beim Positions- und Gierwinkelfehler etwas bessere Ergebnisse als in der thermalen Domäne. Dies spiegelt auch das Ergebnis aus Tabelle 6.2 wider.

Evaluation in schlechten Sichtbedingungen

Schließlich erfolgt die Evaluation aller Ansätze mit sechs Datensätzen in schlechten Sichtbedingungen. Diese wurde entweder durch Aufnahme bei Nacht (*Gym-dark*, *Indoor-dark*, *Outdoor-dark*) oder durch den Einsatz einer Nebelmaschine (*Gym-fog*, *Fog-flight 1*, *Fog-flight 2*) erzeugt. Die letzten beiden Datensätze (*Fog-flight 1*, *Fog-flight 2*) wurden dabei mit der Flugplattform aufgezeichnet. Bei den Datensätzen wurde wieder auf die Abdeckung breiter Anwendungsszenarien geachtet. So werden sowohl Innenräumen (*Gym-dark*, *Indoor-dark*, *Fog-flight 1*, *Fog-flight 2*) also auch urbane Gebiete (*Outdoor-dark*) abgedeckt.

Bei diesen Datensätzen liegt keine Referenz vor. Die Erstellung einer Referenz ist mit dem aktuellen Stand der Technik noch nicht möglich. Stattdessen wird hier der relative finale Positionsfehlerbetrag evaluiert. Da die Start- und Endposition bei allen Datensätzen gleich ist, kann der finale Positionsfehler leicht berechnet werden. Der Betrag davon wird anschließend durch die Trajektorienlänge geteilt umso den relativen Fehler zu erhalten. Da es sich bei allen Verfahren um Odometrie-Ansätze handelt, wächst der Fehler mit dem zurückgelegten Weg. Der relative Fehler berücksichtigt dies und wird daher in der Literatur häufig verwendet.

Visuell-inertiale Ansätze divergieren bei Dunkelheit und erzielen auch schlechtere Ergebnisse bei Sichteinschränkungen durch Nebel. So kann VINS nur zwei und ROVIO lediglich drei der sechs Datensätze in dieser Domäne prozessieren. RRxIO, dagegen, liefert konsistent hohe Genauigkeiten auf allen sechs Datensätzen.

In der thermal-inertialen Domäne erzielt ROVIO deutlich bessere Ergebnisse als VINS. So kann ROVIO zumindest fünf der sechs, VINS allerdings nur einen der sechs Datensätze erfolgreich verarbeiten. RRxIO, dagegen, kann alle sechs Datensätze erfolgreich verarbeiten und erzielt im Schnitt bessere Ergebnisse als ROVIO.

Es werden ähnliche Genauigkeiten sowohl für visuelle als auch thermale Bilder mit RRxIO erzielt (0,68 % versus 0,67 % bei $N = 25$ Features). Die Anzahl an verwendeten Features zeigt wiederum den schon zuvor beobachteten Verlauf. So erzielt RRxIO mit visuellen Bildern einen um 32 % besseren Positionsfehler bei $N = 25$ versus $N = 10$ Features. Bei thermalen Bilder erzielt RRxIO

mit $N = 25$ Features um 34 % bessere Positionsgenauigkeit als bei $N = 10$ Features. Dies entspricht ungefähr der Beobachtung bei den Datensätzen mit Referenz sowohl für visuell (37 % versus 32 %) als auch thermal (32 % versus 34 %).

RRxIO erzielt auch bei diesen Datensätzen bessere Ergebnisse als die reine Radar-inertiale Navigation. So ist die erreichte Positionsgenauigkeit bei RRxIO beider Domänen um circa 12 % besser.

Evaluation Laufzeit

Die beiden unteren Zeilen von Tabelle 6.2 analysieren die Echtzeitfaktor aller Ansätze im Durchschnitt über alle Datensätze. Dafür wird die Zeitdauer des Datensatzes durch die Zeitdauer zum Prozessieren des gesamten Datensatzes geteilt. Ein Vergleich der beiden Embedded-Computer, siehe Unterabschnitt 3.1.2, wurde ebenfalls durchgeführt. VINS erreichte dabei nur einen Echtzeitfaktor von eins auf dem *Intel-NUC-i7* und konnten mit dem *Up-Core* nicht in Echtzeit verarbeitet werden. ROVIO kann deutlich schneller verarbeitet werden und erreicht somit Echtzeitfaktoren von sechs sowie 0,6 auf dem *Intel-NUC-i7* und *Up-Core*. Es zeigt sich, dass x-RIO mit Abstand die schnellsten Laufzeiten erreicht. So erzielt x-RIO einen Echtzeitfaktor von 140 auf dem *Intel-NUC-i7* und 21 auf dem *Up-Core*). RRxIO mit $N = 10$ Features erreicht in beiden Domänen mit dem *Intel-NUC-i7* Laufzeiten 12-Mal schneller als Echtzeit und auch mit dem *Up-Core* 20 % schneller als Echtzeit. Damit erzielt diese Konfiguration von RRxIO circa doppelt so schnelle Echtzeitfaktoren wie ROVIO und mehr als 12-fach schnellere als VINS. Im Gegensatz zu VINS und ROVIO erreicht RRxIO sogar auf dem *Up-Core* Laufzeiten schneller als Echtzeit und kann daher zur online Navigation eingesetzt werden.

6.1.6 Fazit

In diesem Abschnitt wurde die Erweiterung von Kamera-basierten Ansätzen mit Radar-Eigengeschwindigkeitsmessungen präsentiert. Dabei hat sich gezeigt, dass die Fusion von Radar-Eigengeschwindigkeit einen effektiven Weg darstellt, um monokulare VIO- oder TIO-Ansätze zu erweitern. Auf Basis eines etablierten VIO-Ansatzes wurde die Pipeline *robuste Radar visuelle/thermale-inertiale Odometrie* (RRxIO) entwickelt. Hierbei kann RRxIO im Vergleich zum ursprünglichen Ansatz ROVIO in zwei Richtungen ausgelegt werden. Bei gleicher Anzahl an getrackten Features verbessert RRxIO die Positionsgenauigkeit erheblich. Bei einer Verringerung der Anzahl an Features wird immernoch die gleiche Genauigkeit wie von ROVIO erreicht, die Laufzeit verbessert sich jedoch stark.

RRxIO in der visuellen Domäne erzielt sogar bessere Ergebnisse als VINS wobei die Laufzeit von RRxIO mehr als zehn Mal schneller ist. Durch die Fusion mit der Radar-Eigengeschwindigkeit können Phasen mit schlechten oder gar keinen visuellen Features, die zur Divergenz von reinen VIO-Ansätzen führen, effektiv überbrückt werden.

RRxIO in der thermalen Domäne erzielt deutlich bessere Ergebnisse als ROVIO und VINS. So stellen auch Szene mit geringen Temperaturgradienten oder NUCs der Thermalkamera, die zur Divergenz reiner TIO-Ansätze führen, durch die Fusion mit Radar-Eigengeschwindigkeit kein Problem dar.

6.2 Radar-inertiale Navigation mit globaler Stützung

Anwendungen von UAS, wie zum Beispiel der Transport von Gütern, Such- und Rettungsmissionen, Überwachung oder Wartung benötigen ein globales Navigationssystem, um global geplante Missionen abfliegen zu können. Jedoch muss dieses Navigationssystem die globale Position, Geschwindigkeit und Lage des Fluggeräts auch in Umgebungen mit schlechten GNSS-Bedingungen oder gar beim Ausfall von GNSS robust und sicher bereitstellen können. Zudem dürfen auch schwierige Sichtbedingungen wie Dunkelheit, Rauch oder Nebel die Genauigkeit des Navigationssystems nicht beeinflussen.

In diesem Abschnitt wird die Erweiterung der Radar-inertialen Navigation aus Kapitel 5 mit globaler Stützung mittels GNSS beschrieben. Dieser Ansatz wird im Folgenden GNSS-Radar-inertiale Navigation genannt und wurde in Doer et al. [172] veröffentlicht.

Der Vorteil von GNSS-Radar-inertiale Navigation liegt in der Robustheit gegenüber schlechten Sichtbedingungen und profitiert von den komplementären Eigenschaften von Radar und GNSS-Navigation. Während die Radarnavigation Strukturen innerhalb des Sichtbereichs des Radars benötigt, sind die GNSS-Bedingungen am besten, wenn keine Strukturen, wie zum Beispiel Gebäude, in der Nähe sind (Open Sky) [172]. Die Kombination aus Radar und GNSS stellt somit ein vielversprechendes Navigationssystem dar, dass sowohl in guten als auch, wenn initialisiert, in Umgebungen ohne GNSS eine globale Navigationslösung bereitstellt. Dies ermöglicht Anwendungen wie autonome Flüge innerhalb der vielfältigen Einsatzorte, die sich aus den unterschiedlichen Anwendungen ergeben. So können Missionen, wie beispielsweise Wegpunktflüge, mit globalen Wegpunkten einfach geplant und durchgeführt werden. Das vorgestellte System ist sehr effizient hinsichtlich der erforderlichen Rechenleistung, was eine globale Navigation auch von kleinen UAS mit begrenzten Ressourcen des Embedded-Computers ermöglicht.

Im Folgenden werden zunächst verwandte Arbeiten vorgestellt, dann die Erweiterungen zur globalen Stützung erläutert und anschließend eine ausführliche Evaluation durchgeführt.

6.2.1 Verwandte Arbeiten

Die Navigation in schwierigen GNSS-Bedingungen ist Gegenstand aktueller Forschung. Schleicher et al. [130] kombinieren einen visuellen SLAM Ansatz mit horizontalen GPS-Positionsmessungen. GPS-Messungen sowie Relativmessungen aus einem visuellen SLAM werden auch in [89] mit einem Error State EKF fusioniert. Eine Graphen-basierte Optimierung aus visueller-inertialer Odometrie und GPS-Messungen wird dagegen in [93] präsentiert. Die Navigation eines UAVs wird von Shen et.al [134] mit Hilfe der Fusion einer Kamera, eines Laserscanners sowie GPS-Messungen mit einem Unscented-Kalman-Filter realisiert. All diese Ansätze benötigen jedoch eine visuelle Kamera zur Stützung in Umgebungen mit schlechten GNSS-Bedingungen oder dem Ausfall von GNSS. Somit stellen sie keine Lösung für die entsprechenden Szenarien mit schlechten Sichtbedingungen, wie beispielsweise Dunkelheit, dar.

Die Fusion eines Laserscanners mit GNSS wird in den Arbeiten [84, 94, 99] mit Kalman-Filter-basierten Ansätzen betrachtet. Während Meng et al. [94] Sprünge auf Grund GNSS-Positionsmessungen geringer Qualität durch Relativposen mittels eines Laserscanners vermindert, erfolgt ein hartes Umschalten zwischen GNSS- und Laser-basierter Navigation in [84]. Im Ansatz von Müller et al. [99] wird dieses harte Umschalten durch eine adaptive Fusion vermieden. GNSS-Messungen werden gemäß ihrer Qualität klassifiziert und basierend darauf die Fusion angepasst. So wird bei sehr guten GNSS-Bedingungen keine Laserstützung durchgeführt. Bei mittleren sowie schlechten wird dagegen eine Fusion von Laser-basierten Relativposen durchgeführt wobei das Messrauschen der Laserstützung adaptiv an die Bedingungen angepasst wird. Obwohl die betrachteten Laser-basierten Ansätze [84, 94, 99] vielversprechende Ergebnisse liefern und wegen des Laserscanners auch in Dunkelheit eingesetzt werden können, kann das Szenario der erschwerten Sichtbedingungen durch Rauch oder Nebel, siehe Abschnitt 3.1.1, nicht gelöst werden.

Die Verwendung von Radar löst dagegen die Herausforderungen durch schlechte Sichtbedingungen, siehe Abschnitt 3.1.1. Wie bereits in den verwandten Arbeiten der Radar-inertialen Navigation, siehe Abschnitt 5.1, analysiert, wurde die Fusion von Radar und Inertialensorik zur Navigation in Umgebungen ohne GNSS in den letzten Jahren aktiv erforscht. Die Kombination mit GNSS-Messungen wurde hingegen bisher nicht ausreichend untersucht. Zwar nutzen

Mostafa et al. [97] auch Radarsensorik allerdings nur zur eindimensionalen Geschwindigkeitsschätzung und nicht zur Stützung der dreidimensionalen Geschwindigkeit. So ist eine Stützung der Navigationslösung in allen drei Dimensionen mit dem System nach [97] nicht möglich. Diese Lücke wird mit der Erweiterung der Radar-inertialen Navigation aus Kapitel 5 mit der Fusion von GNSS geschlossen. Das resultierende System wurde in Doer et al. [172] veröffentlicht. Dabei wird ähnlich wie in Müller et al. [99] eine Klassifikation der GNSS-Bedingungen durchgeführt und die Fusion ohne hartes Umschalten umgesetzt.

6.2.2 GNSS-Radar-inertiale Navigation

Abbildung 6.6 zeigt das Blockdiagramm der GNSS-Radar-inertiale Navigation. So wird die Radar-inertiale Navigation, siehe Abbildung 5.1, um die Komponenten GNSS-Empfänger, GNSS-Zustandserweiterung, GNSS-

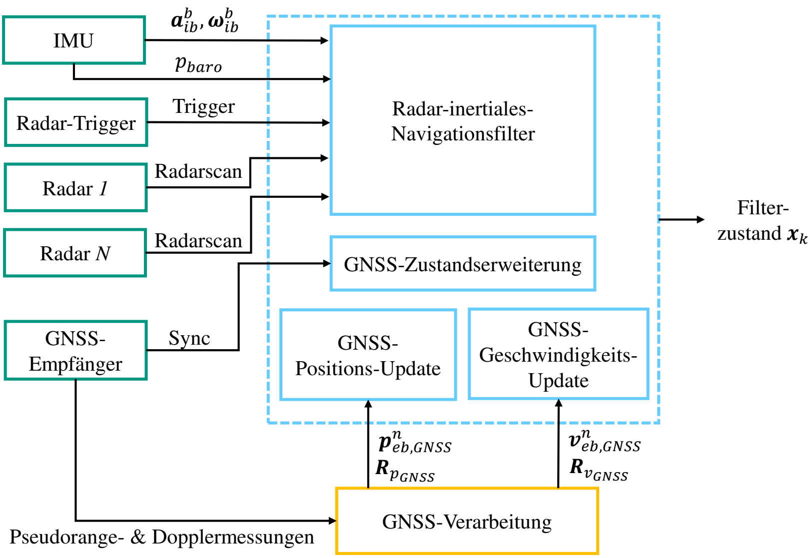


Abbildung 6.6: Blockdiagramm des GNSS-Radar-inertiale Navigationsfilters

Verarbeitung sowie die GNSS Filter Updates erweitert. Da bei den GNSS-Messungen Verzögerungen auftreten bis diese der GNSS-Verarbeitung vorliegen, wird eine Zustandserweiterung mit Hilfe des Sync-Signals eingesetzt. Das Sync-Signal gibt dabei die Gültigkeit der GNSS-Messung vor dem eigentlichen Eintreffen dieser mit vernachlässigbarer Latenz an. Aus den GNSS-Rohdaten der Pseudorange- und Dopplermessungen werden die GNSS-Positionslösung $\boldsymbol{p}^n_{\mathrm{eb,GNSS}}$ und Geschwindigkeitslösung $\boldsymbol{v}^n_{\mathrm{eb,GNSS}}$ berechnet und mit dem ermittelten Messrauschen $\mathbf{R}_p, \mathbf{R}_v$ dem Filter zugeführt.

GNSS-Verarbeitung

Die Verarbeitung der Messdaten des GNSS-Empfängers erfolgt wie in Müller et al. [99] beschrieben. Pseudorange-Messungen werden zunächst gemäß der GPS-Spezifikation [2] korrigiert. Wie in [53, 64] vorgeschlagen werden die Position und Geschwindigkeit mithilfe eines gewichteten Least-Squares Ansatzes berechnet. Hierbei werden die Gewichte w für jede Pseudorange- und Doppler-Messung mit dem Model nach Kuusniemi [78] aufgestellt. Um GNSS-Messungen von Satelliten niedriger Qualität von vorneherein auszuschließen, werden Messungen mit einem C/N_0 unter 33 dB Hz und einem Elevationswinkel unter 15° herausgefiltert [99]. Die Varianzen der Position und Geschwindigkeit werden proportional zu dem PDOP bestimmt [78]. Messungen mit einer großen Varianz deuten auf schlechte GNSS-Bedingungen hin und werden daher direkt ausgeschlossen.

Im Anschluss wird eine Filterung über ein gleitendes Fenster von GNSS-Positionsmessungen durchgeführt [172]. Das Fenster wird dabei zeitlich mit $t \in [t_0, t_0 + T]$ als auch örtlich mit $d \in [0, D]$ definiert. Dabei beschreibt t_0 den Zeitpunkt und d die akkumulierte Strecke relativ zur ersten Messung innerhalb des Fensters. Damit werden die aktuell vorherrschenden GNSS-Bedingungen in gut oder schlecht und folglich die Messungen als gültig oder ungültig klassifiziert. Innerhalb des Fensters bestehend aus N Messungen wird ein Gütemaß j berechnet

$$j = \frac{1}{N} \sum_{i}^{N} \sigma_{p,i}, \tag{6.7}$$

wobei die geschätzte Positionsvarianz σ_p als Gütekriterium dient. Liegt das Gütemaß j oberhalb einer Schwelle wird die GNSS-Messung als gültig klassifiziert und in den beiden Filter Updates verarbeitet. Da das Navigationsfilter durch Radarstützung auch über lange Sicht (z.B. 200 s) nur sehr geringe Positionsfehler akkumuliert, wird die Schwelle j sehr strikt gewählt. Damit wird sichergestellt, dass nur GNSS-Messungen mit sehr hoher Güte fusioniert werden. Wie sich bei den experimentellen Ergebnissen gezeigt hat, werden verhältnismäßig wenige gültige GNSS-Messungen für eine globale Navigationslösung hoher Genauigkeit benötigt.

Filterzustand

Das Radar-inertiale Navigationssfilter aus Abschnitt 5.4 schätzt die Navigationslösung im lokalen Koordinatensystem, siehe Filterzustand (5.18). Die in diesem Abschnitt vorgestellte Erweiterung, zielt jedoch auf ein globales Navigationsfilter ab. Dementsprechend wird der IMU-Zustand (5.18) mit

$$\boldsymbol{x}_{k,\mathrm{IMU}} = \begin{bmatrix} \boldsymbol{p}_{eb}^{n,T} & \boldsymbol{v}_{eb}^{n,T} & \boldsymbol{q}_{b}^{n,T} & \boldsymbol{b}_{a}^{b,T} & \boldsymbol{b}_{g}^{b,T} & b_b \end{bmatrix}^T \tag{6.8}$$

neu definiert und schätzt damit die globale Navigationslösung. Diese umfasst die Translation sowie Geschwindigkeit des körperfesten- relativ zum erdfesten-Koordinatensystem ausgedrückt im Navigationskoordinatensystem. Die Lage beschreibt analog dazu die Rotation vom körperfesten- ins Navigationskoordinatensystem. Da das Navigationskoordinatensystem in Gegensatz zum lokalen-Koordinatensystem in der horizontalen Ebene mit Nord und Ost fest definierte Richtungen hat sowie das erdfeste-Koordinatensystem einen fest definierten Ursprung, ist eine Initialisierung wie in Abschnitt 5.4.1 nicht ausreichend. Der initial Gierwinkel ψ_{nb} kann mittels eines Kompasses initialisiert werden. Alternativ kann ψ_{nb} auch auf Basis von Vorwissen in Form einer groben initialen Ausrichtung Richtung Norden, wie in dieser Arbeit verwendet, mit einer Genauigkeit von circa 20° erfolgen. Die Initialisierung der globalen Position erfolgt über Mittelung der GNSS-Positionsmessungen und benötigt daher gute GNSS-Bedingungen.

Zustandserweiterung

Bei dem eingesetzten Hardwareaufbau können Signalverzögerungen bis zu 70 ms, je nach Anzahl der verfügbaren Satelliten, auftreten. Bei einer Geschwindigkeit von 15 m/s hätte sich in dieser Zeit die Position um circa 1 m geändert. Für eine bestmögliche Sensordatenfusion ist es daher wichtig diese Latenz korrekt abzubilden. Dies wird mit Hilfe des Sync-Signals, dass den korrekten Zeitstempel der GNSS-Messung nahezu frei von Latenz angibt, gelöst. Bei Empfang eines Sync-Signals wird eine Zustandserweiterung, siehe Unterabschnitt 2.2.2, wie folgt ausgeführt:

$$\boldsymbol{x}_{k,\text{erweitert}} = \begin{bmatrix} \boldsymbol{x}_k^T & \boldsymbol{x}_{k,\text{GNSS}}^T \end{bmatrix}^T, \tag{6.9}$$

$$\boldsymbol{x}_{k,\text{GNSS}} = \begin{bmatrix} \boldsymbol{p}_{eb,k}^{n,T} & \boldsymbol{v}_{eb,k}^{n,T} \end{bmatrix}^T. \tag{6.10}$$

Analog wird die Kovarianz erweitert und der Klon aus dem Zustand entfernt sobald das Filter-Update durchgeführt wurde.

GNSS-Filter-Updates

Die GNSS-Filter-Updates fusionieren die Positionsmessung $\boldsymbol{p}_{eb,\text{GNSS}}^n$ sowie die Geschwindigkeitsmessung $\boldsymbol{v}_{eb,\text{GNSS}}^n$, die von der GNSS-Verarbeitung berechnet wurden. Die ebenfalls geschätzte Unsicherheit für die Positions- und Geschwindigkeitsmessung wird als Messrauschen im Filter-Update genutzt. Das GNSS-Filter-Update wird für die Position mit dem Residuum $\boldsymbol{r}_p$

$$\boldsymbol{r}_p = \hat{\boldsymbol{p}}_{eb}^n - \boldsymbol{p}_{eb,\text{GNSS}}^n \tag{6.11}$$

und für die Geschwindigkeit mit dem Residuum $\boldsymbol{r}_v$

$$\boldsymbol{r}_v = \hat{\boldsymbol{v}}_{eb}^n - \boldsymbol{v}_{eb,\text{GNSS}}^n \tag{6.12}$$

modelliert. Ein Mahalanobis-Distanz-Test, siehe Abschnitt A.1, stellt dabei sicher, dass keine Outlier fusioniert werden.

Gierwinkelstützung

Insbesondere in urbanen Umgebungen hat sich die Manhattan-Welt Annahme lokal als anwendbar herausgestellt. So können auch sehr lange Abschnitte ohne GNSS-Stützung kompensiert werden und die Drift des Gierwinkels und daraus folgend der Position erheblich vermindert werden. Bei Realtests hat sich für den Einsatz der Radar-Gierwinkelstützung aus Abschnitt 5.3 als sehr vorteilhaft herausgestellt keine Radar-Gierwinkelstützung bei guten GNSS-Bedingungen durchzuführen. Liegt ein globaler Gierwinkelfehler vor, kann dieser mit Hilfe der GNSS-Messungen der Position und Geschwindigkeit und translatorischer Bewegung beobachtet und korrigiert werden. Dies würde aber im Konflikt zur lokalen Radar-Gierwinkelstützung stehen, da die Korrektur des globalen Fehlers diesem widersprechen würde. Daher erfolgt bei guten GNSS-Bedingungen keine relative Gierwinkelstützung auf Basis der Radarsensorik sondern eine absolute auf Basis von GNSS-Messungen. Da die Manhattan-Welt Annahme auch in urbanen Umgebungen meist nur lokal gültig ist, erfolgt beim Wechsel von guten zu schlechten GNSS-Bedingungen eine Reinitialisierung des Manhattan-Welt Winkels gemäß Abschnitt 5.3.3. Nach erfolgreicher Initialisierung wird die Gierwinkelstützung, wie in Abschnitt 5.3.4 und 5.4.3 präsentiert, ausgeführt.

6.2.3 Ergebnisse

Experimentelle Ergebnisse sind mit dem Hardwareaufbau, wie in Kapitel 3 beschrieben, sowohl für getragene als auch geflogenen Datensätze evaluiert. Hierbei liegt der Fokus auf urbanen Umgebungen, welche herausfordernde GNSS-Bedingungen insbesondere in Gebäudenähe aufweisen. Auf Grund von Effekten wie beispielsweise Mehrwegeausbreitung und Abschattung von Satellitensignalen, können hier sehr große GNSS Fehler auftreten auf. Die *KIT-Campus-Datensätze* verdeutlichen hierbei die Performance von GNSS-Radar-inertialer Navigation auf urbanen Datensätzen die einen großen Bereich abdecken. Aus rechtlichen Gründen konnten hierbei nur getragene Datensätze aufgezeichnet werden. Dadurch konnten jedoch komplexere Szenarien abgedeckt werden als durch Flugdatensätze und durch die Wahl des gelaufenen Pfades eine nachträgliche quantitative Evaluation durchgeführt werden. Die *DRZ-Datensätze* demonstrieren den Ansatz in industrieller Umgebung sowohl

mit getragenen als auch geflogenen Datensätzen und bestätigen die Ergebnisse der *KIT-Campus-Datensätze* in Flugexperimenten. Es wird bei allen Datensätzen lediglich GPS ohne die Hinzunahme von DGPS, SBAS oder anderweitigen Korrekturdiensten eingesetzt.

KIT-Campus-Datensätze

Abbildung 6.7 zeigt die Ergebnisse aller drei *KIT-Campus-Datensätze* auf einem Satellitenbild. Das Navigationsfilter wurde dabei ohne vorheriges Wissen der globalen Position nur mit Hilfe der GPS-Messungen initialisiert. Trotz der Verwendung von Standard-GPS (SPS) und der schwierigen GNSS-Bedingungen liefert das Navigationssystem durchgehend sehr glatte Trajektorien ohne Sprünge der horizontalen Position selbst beim Wechsel von schlechten zu guten GNSS-Bedingungen. Dies gelingt vor allem auch durch die Dopplermessungen. Markierung 4 in Abbildung 6.7 zeigt beispielsweise zwischen den beiden Hochhäusern solche Wechsel. Die geschätzte entspricht der eigentlichen Trajektorie und zeigt, dass fehlerhafte GPS-Messungen korrekt ausgeschlossen werden. Aus rechtlichen Gründen wurden die Datensätze mit der getragenen Multisensorplattform aufgezeichnet. Dies ermöglicht wiederum eine geeignete Wahl der Trajektorie, bei der klar erkennbare Strukturen wie beispielsweise Gullydeckel, Randsteine und Eingängen jeweils mittig überschritten wurden. So kann eine quantitative Evaluation an diesen markanten Punkten, in Abbildung 6.7 durch die nummerierten Marker gekennzeichnet, durchgeführt werden. Vergrößerte Ausschnitte an Markierungen 2, 9, 11 und 14 ist in Abbildung 6.8 gezeigt. Dabei ist der approximative Referenzpfad in Rot eingezeichnet. Auch an diesen herausfordernden Stellen ist der globale Positionsfehler recht klein. So wird ein Maximum von 2,2 m bei Markierung 11 erreicht.

Eine Zusammenfassung der quantitativen Evaluation aller Markierungen ist in Tabelle 6.3 aufgeführt. Hierbei ist der horizontale Positionsfehler evaluiert, indem die Distanz der geschätzten Trajektorien mit dem approximativen Referenzpfad an den markanten Punkten bestimmt wurde. Dies wurde an insgesamt 27 Punkten durchgeführt. Dabei wird angenommen, dass das Satellitenbild global sehr genau referenziert ist. Als Grundlage ist hier das Tool *Google-Earth-Pro* eingesetzt worden, was in dieser Arbeit als Referenz ausreichender Genauigkeit eingestuft wird. GNSS-Radar-inertiale Navigation

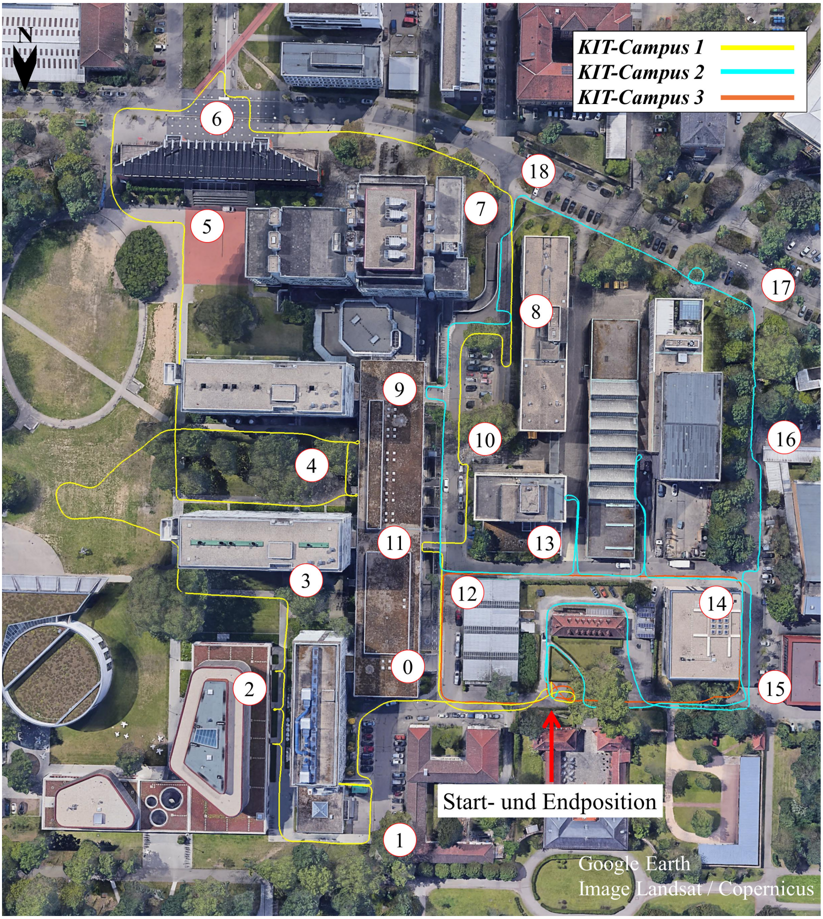


Abbildung 6.7: Geschätzte Trajektorien von allen drei Läufen der *KIT-Campus*-Datensätze in gelb, cyan und orange auf einem Satellitenbild projiziert, basierend auf Doer et al. [172]. Die Trajektorien zeigen trotz der erschwerten GNSS-Bedingungen einen sehr glatten Verlauf, welcher dem Referenzpfad sehr genau folgt. Der gelaufene Pfad wurde so gewählt, dass klar erkennbare Strukturen wie Bordsteine, Gullydeckel oder Eingänge jeweils zentral überschritten wurden. So kann an markanten Stellen, durch Marker gekennzeichnet, eine quantitative Evaluation durchgeführt werden.

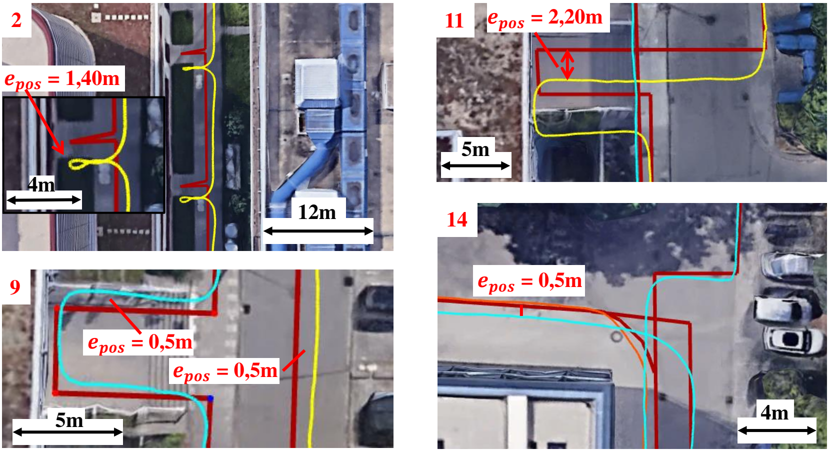


Abbildung 6.8: Vergrößerte Ausschnitte an Markierungen 2, 9, 11 und 14 mit geschätzter Trajektorie und approximativen Referenz in rot, basierend auf Doer et al. [172]. Anhand der eindeutigen Strukturen und des gelaufenen Pfades ist eine quantitativen Evaluation möglich. Satellitenbilder © Google Earth Image Landsat / Copernicus.

KIT-Campus 1	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**	
	1,7	1,4	1,6	1,5	0,3	0,3	1,4	0,6	1,0	0,3	2,2	
KIT-Campus 2	**7**	**8**	**9**	**10**	**11**	**12**	**13**	**14**	**15**	**16**	**17**	**18**
	0,9	0,6	0,5	0,6	0,2	0,2	0,1	0,5	1,5	0,3	3,2	0,1
KIT-Campus 3	**12**	**13**	**14**	**15**								
	0,3	0,1	0,6	1,75								

Tabelle 6.3: Evaluation des globalen horizontalen Positionsfehlers in Metern der *KIT-Campus-Datensätze* an Markierungen 1 - 18 (in fett gekennzeichnet) nach Doer et al. [172].

erzielt in den *KIT-Campus-Datensätzen* einen maximalen horizontalen Positionsfehler von 3,2 m an Markierung 17. Der Mittelwert des horizontalen globalen Positionsfehlers beträgt nur 0,9 m. Dies ist bemerkenswert, da das verwendete Standard-GPS laut Datenblatt [162] bei Open-Sky-Bedingungen und mindestens sechs Satelliten einen horizontalen Streukreisradius (Circular Error Probable - CEP) von 2,5 m aufweist. Die Fusion mit Radar verbessert also die Positionsgenauigkeit deutlich und behält dies auch bei schweren GNSS-Bedingungen bei. Zudem sind die GNSS-Bedingungen bei den gege-

benen Datensätzen nochmals erschwert, da die Plattform nur 1,5 m über dem Boden getragen wurde. Im Vergleich zu höheren Fluglagen ergeben sich hier deutlich schlechtere GNSS-Bedingungen.

Abbildung 6.9 zeigt die GPS-Positionsmessungen der ersten beiden *KIT-Campus-Datensätze*, wobei die als gültig klassifizierten in grün und ungültig klassifizierte in rot eingefärbt sind. Innerhalb guter GNSS-Bedingungen (grün) zeigen sich nur geringe Abweichungen der GPS-Positionsmessungen. In der direkten Nähe zu Gebäuden, durch Markierung *a* gekennzeichnet, tritt eine sehr starke Streuung der GPS-Messungen auf. Diese werden jedoch als ungültig eingestuft und somit nicht fusioniert. Auch der Wechsel von schlechten zu guten GNSS-Bedingungen, siehe Markierung *b*, verläuft sehr glatt. Da die Radar-inertiale Navigation nur geringe Fehler während der schlechten GNSS-Bedingungen akkumuliert hat, tritt an diesen Stellen keine Sprünge in der Position auf. Es werden nur wenige gültige GNSS-Messungen benötigt.

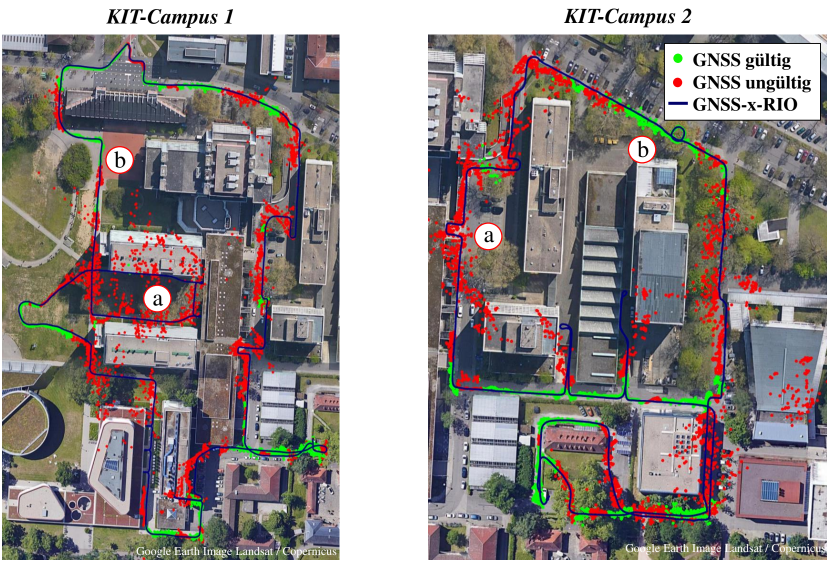


Abbildung 6.9: GPS Fixes der *KIT-Campus*-Datensätze mit Klassifikation in gültig (grün) und ungültig (rot) sowie der geschätzten Trajektorien (schwarz) nach Doer et al. [172]. Durch die Klassifikation werden ungültige Messungen effektiv ausgeschlossen.

DRZ-Datensätze

Die folgenden Datensätze wurden am Deutschen Rettungsrobotik-Zentrum (DRZ) aufgezeichnet. Hierbei wurde der Übergang von guten zu schlechten GNSS-Bedingungen betrachtet, zunächst mit getragenen und anschließend mit geflogenen Datensätzen.

Das Ergebnis der drei getragenen Datensätze ist in Abbildung 6.10 dargestellt. Die Startphase ist so gewählt, dass gute GNSS-Bedingungen für eine gute globale Initialisierung des Navigationsfilters vorliegen. Anschließend wird das Gebäude betreten, wobei das Eingangstor genau mittig durchschritten wurde. Beim Datensatz *DRZ-getragen 2* wurde zusätzlich der Pfad im rechten unteren Bereich genau mittig zwischen den beiden Gebäuden gewählt. Die vergrößerte Darstellung, siehe rechts oben in Abbildung 6.10, zeigt, dass die geschätzte globale Position sehr genau zum Satellitenbild passt. Die maximale globale horizontale Abweichung zur Mitte des Eingangstors ist lediglich 0,7 m und wurde mit dem Tool *Google Earth Pro* bestimmt. Der Übergang erfolgt ohne Sprünge der Position und zeigt damit einen sehr glatten Verlauf. Es werden also fehlerhafte GNSS-Messungen in der Nähe des Gebäudes effektiv ausgeschlos-

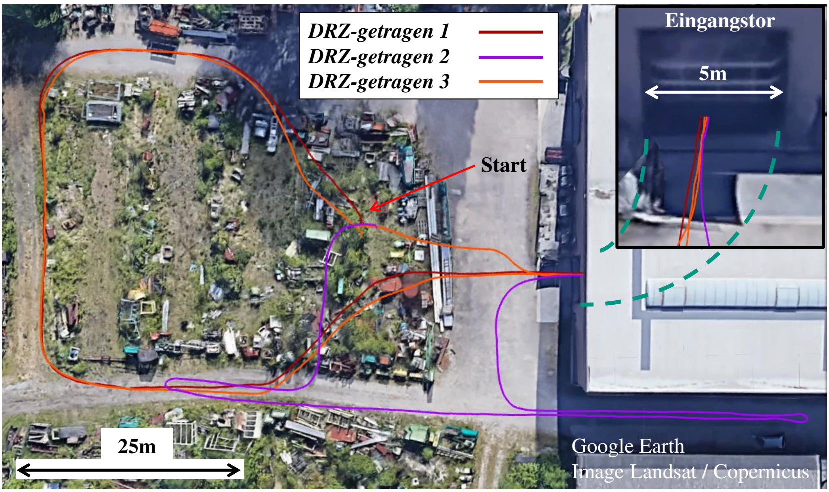


Abbildung 6.10: Getragene DRZ-Datensätze mit Satellitenbild nach Doer et al. [172]

sen. Dabei wird eine hohe Genauigkeit der globalen Positionslösung erzielt. Der sehr ähnliche Verlauf der drei Trajektorien beim Durchschreiten des Tors deutet auf ein robustes Systems hin.

Die Evaluation durch drei manuelle Testflüge ist in Abbildung 6.11 dargestellt. Analog zu den getragenen Datensätzen erfolgt zunächst die Initialisierung in guten GNSS-Bedingungen. Darauf folgt der Flug in das Gebäude mittig durch das Eingangstor. Auf Grund von starkem Wind, konnte dabei ein glatter Pfad, wie bei den getragenen Datensätzen, exakt mittig durch das Tor für den Flug 1 und 3 nicht ausgeführt werden. Eine Auswertung der onboard Kamera sowie externen Kameraaufnahmen ergab, dass die Abweichungen sehr genau zum tatsächlichen Flugpfad passen. Die Testflüge zeigen die gleiche Performance wie die getragenen Datensätze und bestätigen die Anwendbarkeit der GNSS-Radar-inertiale Navigation für die Navigation von UAS. Zusätzlich zeigt sich bei den *DRZ-Datensätzen* auch die hohe Güte der globalen Positionsschätzung, die in diesem Fall ausreichen würde, um ein global geplanten Flug in eine Lagerhalle mit 5 m breiten Eingangstor mit ausreichend Sicherheitsabstand durchzuführen.

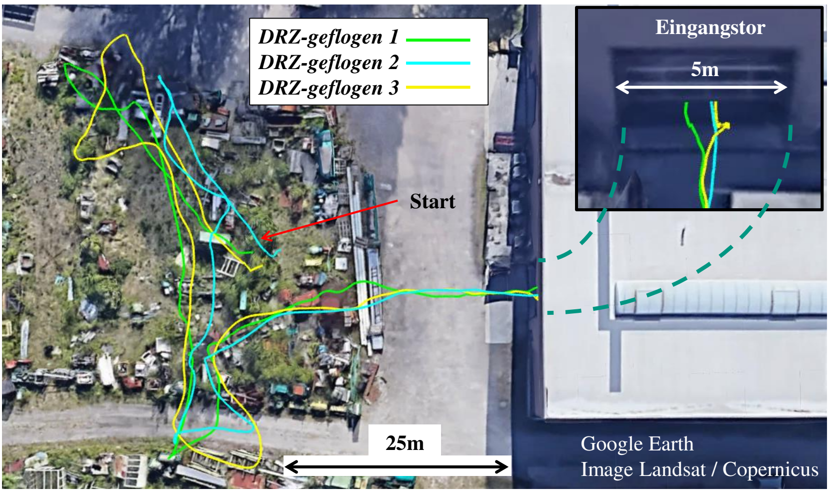


Abbildung 6.11: Gefolgenen DRZ Datensätze mit Satellitenbild nach Doer et al. [172]

6.2.4 Fazit

Mit der Erweiterung der Radar-inertialen Navigation aus Kapitel 5 um die Fusion von GNSS-Messungen konnte ein globales Navigationssystem entworfen werden. Eine Klassifizierung der GNSS-Messungen hinsichtlich gültiger und ungültiger Messungen ermöglicht eine robuste Fusion der GNSS-Positions- und GNSS-Geschwindigkeitsmessungen. Durch die Radarstützung wird auch bei schlechten GNSS-Bedingungen oder bei Ausfall von GNSS die globale Position mit hoher Güte bereitgestellt. Selbst bei Verwendung von Standard-GPS ohne externe Korrekturen durch DGPS, SBAS oder anderweitige Korrekturdienste wird ein horizontaler Positionsfehler von maximal 3,2 m und durchschnittlich 0,9 m in schwierigen urbanen Bedingungen erreicht, wie in umfangreichen Datensätzen evaluiert worden ist. Somit können UAS-Anwendungen, wie beispielsweise der Transport von Personen oder Gütern, Such- und Rettungsmissionen, Überwachung oder Wartung auch in schwierigen Bedingungen realisiert werden. So stellt die GNSS-Radar-inertiale Navigation auch bei schlechten Sichtbedingungen, wie beispielsweise bei Rauch, Nebel oder Dunkelheit, eine globale Navigationslösung bereit. Dadurch werden Anwendungen möglich, die mit dem aktuellen Stand der Technik nicht realisierbar sind. Zusätzlich ist das entworfene Navigationsfilter hinsichtlich der benötigten Rechenleistung sehr effizient. Mit dem Intel-NUC-i7 werden Laufzeiten hundertmal schneller als Echtzeit und auf dem *Up-Core* Laufzeiten fünfzehnmal schneller als Echtzeit erreicht. Bei Verwendung des *Up-Cores* wird somit auch die Verwendung für klein-UAS ermöglicht.

7 Kartierung in schwierigen Sichtbedingungen

In diesem Kapitel wird die bisher unveröffentlichte Pipeline zur Kartierung der Umgebung selbst bei schlechten Sichtbedingungen präsentiert. Diese kombiniert die Daten der Radarsensoren sowie der Thermalkamera zu einer multisensoriellen Kartierung. Hierbei wird die Radar-inertiale Navigation um Simultanous Localization and Mapping (SLAM) erweitert. Dadurch wird eine langzeitstabile Positionsschätzung erreicht, was durch die Fähigkeit Schleifenschlüssen zu detektieren realisiert wird. Zusätzlich erfolgt die Detektion und Klassifikation der Körperhaltung hinsichtlich *aufrecht*, *sitzend*, *liegend* oder *unbekannt* mit Hilfe neuronaler Netze sowie die Kartierung mit Hilfe des SLAMs. Mit dem Ziel einer Kartierung während des Flugs liegt der Fokus bei der Konzeption und Implementierung auf Echtzeitfähigkeit. Abbildung 7.1 fasst die wichtigsten Bausteine der Pipeline zusammen und besteht aus folgenden vier Hauptblöcken:

- Radar-inertiales Navigationsfilter: Das in Abschnitt 5.4 vorgestellte Navigationsfilter stellt den kombinierten Radarscan sowie die Navigationslösung für den eingeführten Radar-SLAM und die Radar-Kalibrierung für die Personen-Kartierung bereit. Der kombinierte Radarscan fasst die aktuellsten Inlier-Radarscans, die Scans aller statischen Elemente, von allen Radarsensoren transformiert ins körperfeste Koordinatensystem zusammen.

- Radar-SLAM: Zentrale SLAM-Pipeline, die zunächst eine Scan-Filterung des kombinierten Radarscans durchführt und diesen anschließend mit dem etablierten Laser-SLAM-Verfahren *Cartographer* [55] verarbeitet, was die SLAM-Trajektorie sowie die SLAM-Rasterkarte der Umgebung liefert. Auf Grund der Scan-Filterung kann *Cartographer* Schleifenschlüsse effektiv auch mit Radarscans detektieren und führt auf Basis davon eine Optimierung der Trajektorie sowie der Karte durch.

- Personen Detektion und Klassifizierung: Zunächst wird das Bild mit Hilfe der Navigationslösung stabilisiert und Linsenverzerrungen entfernt. Anschließend erfolgt die Detektion von Personen sowie der Posenschätzung mit Hilfe von Convolutional Neural Networks (CNNs). Auf Basis der Posenschätzung wird schließlich die Klassifizierung bezüglich der Körperhaltungen *aufrecht*, *sitzend*, *liegend* oder *unbekannt* mit einem Random-Forest-Ansatz durchgeführt.

- Personen-Kartierung: Finaler Block der Kartierungspipeline, der die Personenmessungen in Form von Bounding-Boxen samt klassifizierter Körperhaltung mit Hilfe der SLAM-Trajektorie sowie Radarscans und Radar-Kalibrierungen kombiniert. Dabei erfolgt die Schätzung der Personen-Position jeweils mit einem separaten Kalman-Filter.

Die einzelnen Blöcke werden in den nachfolgenden Unterkapiteln näher beschrieben und evaluiert. Abschließend erfolgt die Demonstration der Gesamtpipeline mit einem autonomen Flug in dichtem Nebel im Abschnitt 7.4.1.

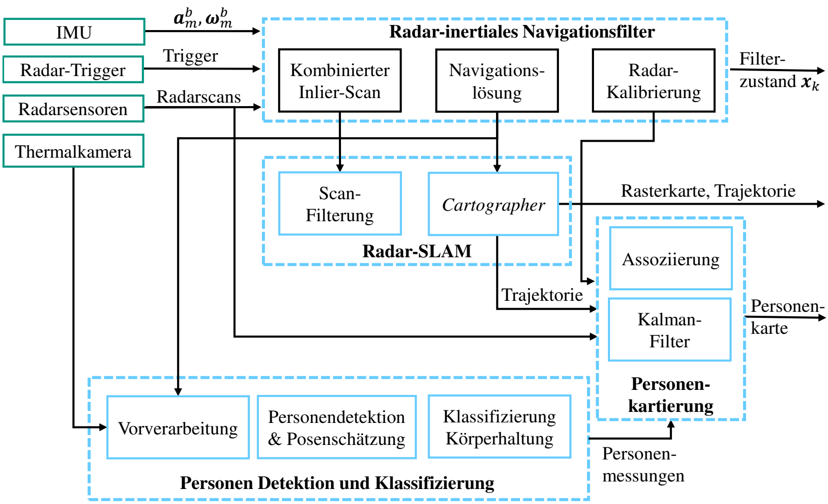


Abbildung 7.1: Blockdiagramm der entwickelten Kartierungspipeline

7.1 Verwandte Arbeiten

SLAM

Laser-SLAM-Verfahren wurden in den letzten zwei Jahrzehnten sehr intensiv erforscht [55, 74, 96, 158] und können daher als sehr ausgereifte Systeme betrachtet werden. Radar-SLAM-Ansätze sind im Gegensatz dazu erst seit kurzem Gegenstand aktueller Forschung [56, 88, 131]. Auf Grund der Ähnlichkeit von Laser- und Radarscans, wurde die Optimierung von FastSLAM2.0 [96] durch Lupfer et al. [88] betrachtet. Dabei wurden die Doppler-Informationen, die bei Radarscans im Gegensatz zu Laserscans vorhanden sind, mit in das Schätzproblem integriert. Schuster et al. [131] lösen dagegen das SLAM-Problem mit Hilfe von Graph-SLAM [142]. Dafür werden robuste Features aus dem Radarscan extrahiert und diese als Landmarken im Graph-SLAM modelliert. Beide Verfahren [88, 131] nutzen jeweils zur Verarbeitung einzelne Radarscans. Der Ansatz Radarscans über ein kurzes Zeitfenster, als sogenannte Submap, zu akkumulieren, um so robusteres Verhalten zu erreichen, wurde in [56] vorgeschlagen. Die Submaps dienen anschließend als Grundlage für Featureextraktion und dem Matching. Um sicherzustellen, dass nur statischen Objekte kartiert werden, werden diese mit einem RANSAC-Verfahren über die Dopplergeschwindigkeit gefiltert [56].

Objektdetektion

Objektdetektion beschreibt die Detektion von Objekten in Bilder in Form einer rechteckigen Bounding-Box mit Hilfe von neuronalen Netzen, sogenannter Convolutional Neural Networks (CNNs), siehe auch Abschnitt 2.4.1 im Grundlagenkapitel. Während die meisten CNNs zur Objektdetektion eine Grafikkarte zur Prozessierung in Echtzeit benötigen [114, 115], wurde in den letzten Jahren an hoch optimierten Netzen für Embedded-Computer oder Smartphones geforscht [125, 141, 157]. Für Realanwendungen werden dabei häufig dedizierte Hardware-Chips eingesetzt und die online Detektion auch auf Embedded-Computern zu ermöglichen. Sehr populär sind dabei sogenannte *Tensor-Processing-Units* (TPUs), die speziell für die Inferenz von *TensorFlow*-Modellen [3] optimiert sind. Dabei wird das CNN zunächst mit Hilfe von Grafikkarten trainiert und anschließend das erstellte *TensorFlow*-Modell

mittels eines Compilers in ein *TensorFlow-Lite*-Modell, ein approximiertes Format, übersetzt. Dieses kann dann auch auf Edge-TPUs verarbeitet werden. Für die etablierten Echtzeit-CNNs *MobileNetV2* [125], *MobileDet* [157] und *EfficientDet* [141] sind durch Google hoch-optimierte CNNs für den *Edge-TPUs* veröffentlicht [26].

Posenschätzung und Klassifizierung

Die Posenschätzung von Personen, siehe auch Abschnitt 2.4.1 im Grundlagenkapitel, ist Gegenstand aktiver Forschung und wurde inzwischen mit Hilfe von CNNs mit hoher Genauigkeit gelöst. Dabei lag der Schwerpunkt auf CNNs, die zwar in Echtzeit die Posenschätzung durchführen können, aber dafür einen leistungsstarken Computer mit Grafikkarte benötigen [19,21,25,77]. Während [19,25,77] visuelle Bilder nutzen, haben Chen et al. [21] die Posenschätzung speziell mit Thermalbildern betrachtet. Analog zur Objektdetektion, zeigt sich auch bei der Posenschätzung der Trend zur Entwicklung von Netzen, die auch mit Embedded-Computern oder Smartphones verwendet werden können [28,106]. So sind für beide Netze auch als Modelle für den *Edge-TPU* exportiert und online verfügbar [27].

Auf Basis der Posenschätzung, wird auch die Schätzung der Körperhaltung in den Veröffentlichungen [50,54] betrachtet. Ghazal et al. [50] setzen dabei ein Regel-basiertes Verfahren zur Unterscheidung von sitzenden und stehenden Posen ein und bauen auf die Posenschätzung *OpenPose* [19] auf. Gupta et al. [54] führten dagegen eine Unterscheidung in die fünf Klassen stehend, sitzend, liegend, rennend und tanzend ein und verglichen verschiedene Verfahren des maschinellen Lernens.

Kartierung von Umgebung und Personen

Die Kartierung der Umgebung auch bei schlechten Sichtbedingungen mittels eines Radarsensors wurde von [85] vorgeschlagen. Dabei wird mit Hilfe eines neuronalen Netzes die Umgebung aus den Radarscans geschätzt und so eine detaillierte Rasterkarte trotz der wenig detaillierten Radarscan erstellt. Zwar entstehen so klare Rasterkarten, darin werden jedoch viele Information interpoliert und die Generalisierung dieses Ansatzes auf untrainierte Umge-

bungen ist fraglich. Zusätzlich wird auch die Kartierung von Personen in der Veröffentlichung [85] nicht betrachtet. Die kombinierte Kartierung der Umgebung und von Personen wird in den Arbeiten [8, 124, 138] präsentiert. Dabei wird ein Laserscanner mit einer Kamera [8, 138] kombiniert oder eine RGBD-Kamera [124] genutzt. Die Kartierung der Umgebung erfolgt mit Hilfe eines SLAMs und die Schätzung der Personenposition basierend auf Kamerabildern. Bei allen drei Ansätzen liegt der Fokus auf guten Sichtbedingungen. Zusätzlich wurden in allen drei Arbeiten lediglich die Detektion von Personen und weiterer Objekte vorgenommen und nicht die Körperhaltung betrachtet.

Fazit

Ziel dieser Arbeit ist eine Kartierung der Umgebung sowie von Personen gemäß dem motivierten Szenario aus Abbildung 1.1 auch in schlechten Sichtbedingungen zu realisieren. Während dies zwar für die Kartierung der Umgebung durch [86] auch in schlechten Sichtbedingungen präsentiert wurde, findet dabei keine Kartierung von Personen statt. Daher wird im Folgenden ein hybrider Ansatz vorgeschlagen, der eine kombinierte Kartierung der Umgebung mittels Radar sowie von Personen mittels Thermalkamera durchführt. Dabei wird das etablierte Laser-SLAM-Verfahren *Cartographer* [55] auf Grund der Performance, Echtzeitfähigkeit und Robustheit sowie der Verwendung von Submaps eingesetzt und für den Einsatz mit Radarscans optimiert. Dies liefert eine detaillierte Rasterkarte der Umgebung sowie eine langezeitgenaue Positionsschätzung durch Schleifenschlüsse in der gleichen Umgebung. Für die Detektion und Schätzung der Pose wird auf moderne CNNs [28, 106, 157] zurückgegriffen und ein Verfahren implementiert, um diese effizient zu nutzen. Die Personenkartierung erfolgt schließlich Kalman-Filter-basiert und kombiniert den SLAM mit der Detektion und Klassifizierung.

7.2 Radar-SLAM

Im folgenden Abschnitt wird die implementierte Radar-SLAM-Pipeline vorgestellt. Dabei handelt es sich um ein unveröffentlichtes Verfahren. Der Radar-SLAM ermöglicht eine langzeitstabile Positionschätzung in der gleichen Umgebung, da durch Schleifenschlüsse eine Drift der Position kompensiert werden

kann. Zusätzlich wird die Umgebung durch den SLAM in Form einer Rasterkarte kartiert, welche für jedes Element des Rastern die Belegungswahrscheinlichkeit modelliert. Der Radar-SLAM wird als zweidimensionales Verfahren implementiert, da so die Rasterkarte einem Grundriss ähnelt und damit eine leicht interpretierbare Repräsentation der Umgebung liefert.

7.2.1 SLAM-Pipeline

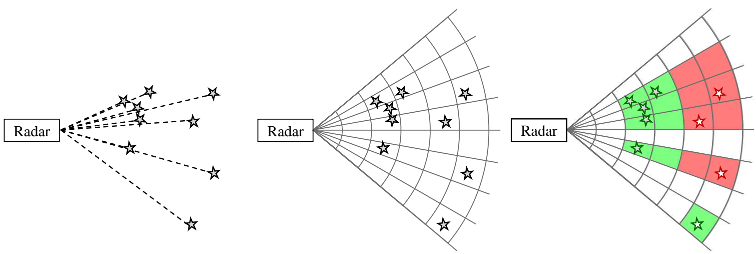
Die SLAM-Pipeline besteht, wie in Abbildung 7.1 dargestellt, aus den Blöcken Scan-Filterung sowie *Cartographer*. Dabei handelt es sich bei *Cartographer* um einen Laser-SLAM, vielseitig einsetzbar und echtzeitfähig ist und daher als SLAM-Pipeline eingesetzt wird. Mit Hilfe der Scan-Filterung und angepasster Parametrisierung ist es möglich *Cartographer* auch mit Radarscans zu verwenden und so von dem ausgereiften SLAM-System zu profitieren.

Cartographer besteht aus einem lokalen und globalen SLAM. Laserscan- und Odometriemessungen werden zunächst zu einer Submap zusammengefasst, wobei neue Scans gegen die bestehende Submap gematcht werden. Die optionale Odometriemessung dient als initialer Startpunkt für das Scanmatching. Da die Submaps nur wenige Scans der direkten Vergangenheit enthalten, akkumulieren sich Fehler. Daher wird eine Optimierung der Posentrajektorie im globalen SLAM durchgeführt. Sobald eine Submap abgeschlossen ist, nach einer definierten Anzahl an Laserscans, wird diese im globalen SLAM für Schleifenschlüsse in Betracht gezogen. Wird ein Schleifenschluss gefunden, wird dieser in Form einer neuen Randbedingung dem Optimierungsproblem hinzugefügt. Der globale SLAM führt die Optimierung nur alle paar Sekunde aus, um die Verarbeitung in Echtzeit zu ermöglichen. [55]

Ziel der Kartierung ist es die statischen Strukturen der Umgebung abzubilden. Daher werden nicht die gemessenen Radarscans, sondern die Inlier-Scans aus der Eigengeschwindigkeitsschätzung von Abschnitt 5.2 genutzt. Diese enthalten nur die statischen Objekte der jeweiligen Szene wobei dynamischen Objekte sowie fehlerhafte Messungen durch den RANSAC-Ansatz entfernt wurden. Zusätzlich hat sich gezeigt, dass *Cartographer* deutlich bessere Ergebnisse erzielt, wenn mindestens zwei Radarsensoren, die einen großen Winkelbereich abdecken, verwendet werden. Daher werden im Radar-inertialen Navigationsfilter die aktuellsten Inlier-Scans aller Radarsensoren in dem kom-

binierten Inlier-Scan zusammengefasst. Um diesen optimal mit *Cartographer* zu nutzen muss noch eine Filterung angewendet werden. Diese basiert auf der Charakteristik der Laserscans, die jeweils pro Richtung oder Strahl nur einen Distanzwert aufweisen. Radarscans, dagegen, sind häufig durch mehrere Scanpunkte mit ähnlicher Richtung gekennzeichnet. Dies gilt insbesondere auf Grund von Reflexionen in Innenräumen. Um dies herauszufiltern, hat sich das in Abbildung 7.2 dargestellte Filterschema bewährt. Der Sichtbereich des Radarscans wird zunächst in Azimut, Elevation und Distanz gerastert, wie in Abbildung 7.2b gezeigt. Dabei wird die Rasterung in Kugelkoordinaten durchgeführt, dass in Abbildung 7.2 für den zweidimensionalen Fall vereinfacht dargestellt ist. Dies spiegelt die Charakteristik des Radarscans wider, da dieser die Distanz und Richtung misst. Anschließend werden für jede Richtung jeweils nur der Scan-Punkt mit der geringsten Distanz als gültig eingestuft, wie in Abbildung 7.2c mit den grünen Sternen dargestellt. Die grünen Bereiche des Rasters markieren dabei gültige Bereiche für weitere Scan-Punkte und die roten Bereich jeweils ungültige.

Im Gegensatz zu Laserscans enthält der Radarscan viel weniger Punkte, daher muss *Cartographer* hierfür gezielt konfiguriert werden. Experimentell hat sich herausgestellt, dass die maximale Anzahl an Radarscans einer Submap verkleinert und die Auflösung der Rasterkarten verringert werden muss. So enthalten Submaps nur maximal 50 Radarscans und anstelle einer Auflösung von 5 cm wird 0,15 cm eingesetzt. Zusätzlich muss die Odometrie im Vergleich zum Scanmatching bei der Optimierung sehr stark gewichtet werden. Das ist zuläs-



(a) Radarscan **(b)** Radarscan mit Rasterung **(c)** Gültige Bereiche

Abbildung 7.2: Veranschaulichung der Radarscan-Filterung mittels einer Rasterung in Distanz und Winkel zweidimensional dargestellt. Die roten Bereiche werde herausgefiltert, da hier jeweils ein näherer Scanpunkt vorhanden ist. Alle Scanpunkte in grünen Bereiche sind gültig.

sig, da die Navigationslösung des Radar-inertialen Navigationsfilters für kurze Zeitspannen sehr genau ist. Außerdem werden dadurch die Ungenauigkeiten des Radarscans gegenüber der erwarteten Laserscans kompensiert. Die Kartierung wird so deutlich robuster, was die langzeitgenaue Positionsschätzung erzielt

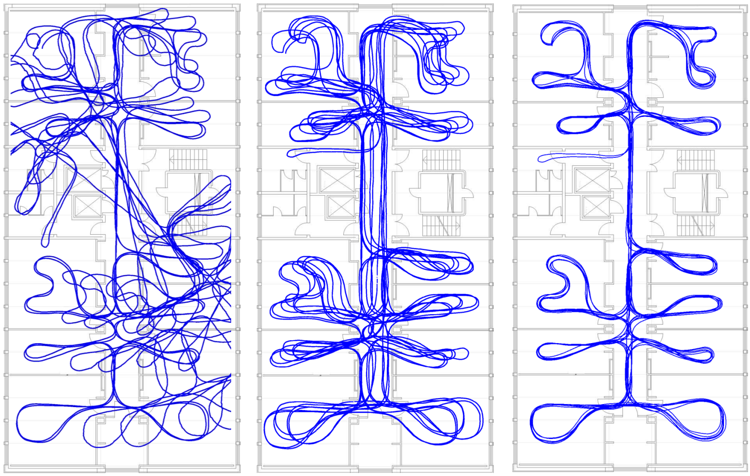
7.2.2 Ergebnisse

In diesem Abschnitt erfolgt zunächst die Präsentation der Ergebnisse des Radar-SLAMs für Innenräume und Außenbereiche auf Basis von getragene als auch geflogene Datensätze. Anschließend werden die Ergebnisse mit denen auf Basis von Laser-Messungen gegenübergestellt.

Radar-SLAM

Der Radar-SLAM wird zunächst mit einem sehr langen Datensatz im Innern eines Bürogebäudes demonstriert. Der Datensatz weist dabei eine Trajektorienlänge von 1290 m über einen Zeitraum von 20 min auf. Es wurden dabei insgesamt sechs Runden mit dem approximativ gleichen Pfad absolviert. Abbildung 7.3 zeigt die Ergebnisse der Radar-inertialen Navigation links und in der Mitte. Dabei zeigt x-RIO ohne Stützung des Gierwinkels, Abbildung 7.3a, einen wachsenden Drift des Gierwinkelfehlers, was zu einem stark wachsenden Fehler der Position führt. Im Gegensatz dazu weist x-RIO mit Radar-Gierwinkelstützung, Abbildung 7.3b, keinen Drift des Gierwinkels auf. Mit der Zeit akkumulieren sich jedoch Fehler der Position. Der implementiere Radar-SLAM, Abbildung 7.3c, zeigt dagegen keinen erkennbaren Drift des Positions- und Gierwinkelfehlers. Somit wird eine langzeit-genaue Positionsschätzung erreicht. Der Radar-SLAM kann dabei effektiv und robust Schleifenschlüsse detektieren und die nachfolgende Optimierung liefert die Driftfreie Trajektorie. Insbesondere bei Anwendungen, die eine Navigation in der gleichen Umgebung erfordern, stellt der Radar-SLAM eine vielversprechende Möglichkeit zur Erweiterung der reinen Radar-inertialen Navigation dar.

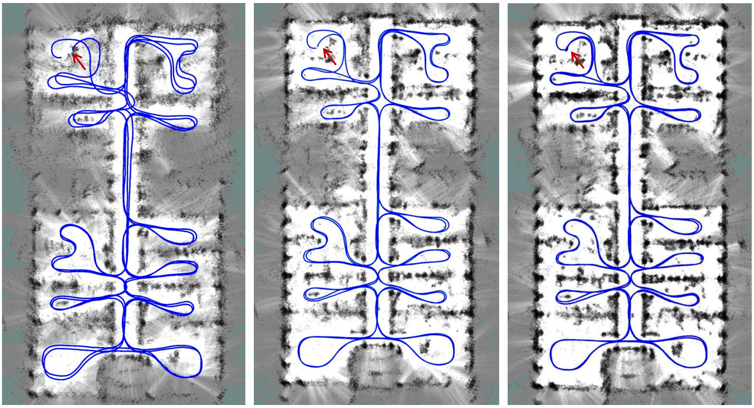
Bei dem Ergebnis aus Abbildung 7.3 wurde der kombinierte Radarscan aller drei Radarsensoren der Multisensor-Plafform aus Abschnitt 3.1 genutzt. Im Folgenden erfolgt die Evaluation hinsichtlich der Radarsensor-Konfiguration



(a) X-RIO **(b)** X-RIO mit Gierstützung **(c)** Radar-SLAM

Abbildung 7.3: Vergleich von x-RIO ohne Stützung (links), x-RIO mit Gierwinkelstützung (Mitte) mit dem Ergebnis des Radar-SLAMs (rechts) in einem Bürogebäude. Der Datensatz hat eine Länge von 1290 m über einen Zeitraum von 20 min und umfasst sechs Runden mit jeweils ähnlichem Pfad. Der Einfluss der Gierwinkelstützung ist deutlich zu erkennen. Der Radar-SLAM liefert langzeitgenaue Positionsschätzungen ohne weitere Drift des Positionsfehlers.

mit dem Datensatz *Office-Floor* aus Abschnitt 5.4.5 der Radar-inertialen Navigation. Abbildung 7.4 zeigt dabei das Ergebnis des Radar-SLAM unter Verwendung von nur dem linken Radarsensor auf der linken Seite, dem linken und rechten Radarsensor in der Mitte sowie aller drei Radarsensoren auf der rechten Seite. Dabei wird sowohl die erstellte Rasterkarte als auch die SLAM-Trajektorie dargestellt. Die Start- und Endposition ist gleich. Es zeigt sich eine Verbesserung des Ergebnisses mit zunehmender Anzahl an Radarsensoren. Besonders deutlich ist dabei die Verbesserung bei Nutzung des linken und rechten im Vergleich zum linken Radarsensor. Durch den umfangreicheren Scan, können Schleifenschlüsse wesentlich besser detektiert werden sowie die Belegungswahrscheinlichkeiten der einzelnen Zellen besser ausgebildet werden. Bei der Konfiguration mit drei Radarsensoren ist eine leichte Verbesserung der Kartierung sowie der Detektion von Schleifenschlüssen zu beobachten. So wird beispielsweise der finale Schleifenschluss, am Ende des Datensatzes, nur mit dieser Konfiguration erreicht.



(a) Ein Radarsensor (l) **(b)** Zwei Radarsensoren (l,r) **(c)** Drei Radarsensoren (l,r,m)

Abbildung 7.4: Vergleich der SLAM Ergebnisse in Abhängigkeit der Anzahl an verwendeten Radarsensoren für den Datensatz *Office-Floor*. Es werden die Konfiguration mit nur dem linken Radarsensor (links), dem linken und rechten Radarsensor (Mitte) und dem linken, rechten und mittleren Radarsensor (rechts) verglichen. Die Qualität des Ergebnisse nimmt mit der Anzahl an Radarsensoren zu.

Abbildung 7.5 vergleicht die erstellte Rasterkarte aus Abbildung 7.4c unter Nutzung von allen drei Radarsensoren mit dem Grundriss des Stockwerkes in rot. Die Rasterkarte und der Grundriss passen sehr gut zusammen und weisen keinen Skalenfehler sowie Drift der Position auf. So ist auch der Raum mit Markierung 1, der am weitesten von den Startposition bei der Markierung 2 entfernt ist, sehr genau kartiert worden. Die Türdurchgänge wurden korrekt kartiert, was die Verwendung der Rasterkarte zum Beispiel für die Einsatzplanung erlaubt. Auch Mobiliar, zum Beispiel bei Markierung 2, konnte als Hindernisse gut abgebildet. Jeweils rechts der Markierung 3 zeigen sich Unterschiede zum Grundriss. Da in diesen Raumecken glatte Wände vorliegen können diese nicht direkt gemessen werden auf Grund der Charakteristik von Radarsensoren an glatten Flächen, wie in Abbildung 5.6 veranschaulicht. Daher werden die Radarwellen hier reflektiert und folglich wird die Wand an diesem Teil dann nicht korrekt kartiert. Auch bei den Markierungen 4 zeigen sich Artefakte auf Grund von Reflexionen. Insgesamt erreicht die Kartierung einen hohen Detailgrad bei gleichzeitiger Robustheit gegenüber schlechten Sichtbedingungen.

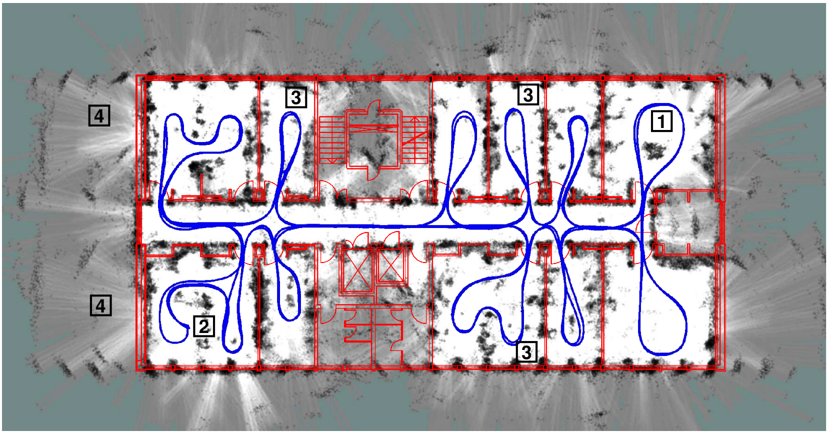


Abbildung 7.5: Evaluation der erstellten Rasterkarte des Radar-SLAMs mit dem Grundriss in rot. Die Rasterkarte sowie der Grundriss passen sehr gut zueinander.

Eine Evaluation des Radar-SLAM ist in Abbildung 7.6 für urbane Umgebungen dargestellt. Dabei wurden wiederum drei Runden des approximativ gleichen Pfades absolviert. Ein Schleifenschluss war auch hier erfolgreich und ein Positionsdrift ist nicht erkennbar. Die Umgebung wurde auch hier sehr gut abgebildet und weist im Vergleich zur Innenraum-Kartierung (*Office-Floor* in Abbildung 7.3) deutlich weniger Reflexionen und Artefakte auf. Markierung 1 zeigt die Kartierung einer glatten Gebäudewand. Markierung 2 deutet auch auf eine Gebäudewand. Diese besteht jedoch auch einer Vielzahl an Fenstern, wie in Abbildung 6.1a oben rechts zu erkennen ist. Markierung 3 zeigt die Reflexion des Baucontainers (Markierung 4) an der Gebäudewand. Auch stehende Fahrzeuge, siehe Markierung 5, sind kartiert worden. Es zeigt sich ein sehr gutes Ergebnis der Kartierung auch von urbanen Außenbereichen.

Abbildung 7.7 demonstriert schließlich die SLAM-Ergebnisse von zwei manuellen Testflügen am Deutschen Rettungsrobtik-Zentrum. Dabei wurde beim *Manuellen Flug* 1 nur eine kleine Schleife geflogen und beim *Manuellen Flug* 2 zwei große Schleifen geflogen. Die Kartierung der etwa $70 \times 30\,\mathrm{m}$ großen Lagerhalle hat auch beim kürzeren Flug sehr gute Ergebnisse geliefert. Auch beim *Manuellen Flug* 2 ist der Schleifenschluss sehr robust detektiert worden und ein Positionsdrift tritt nicht auf. Insgesamt zeigen sich auch bei den beiden

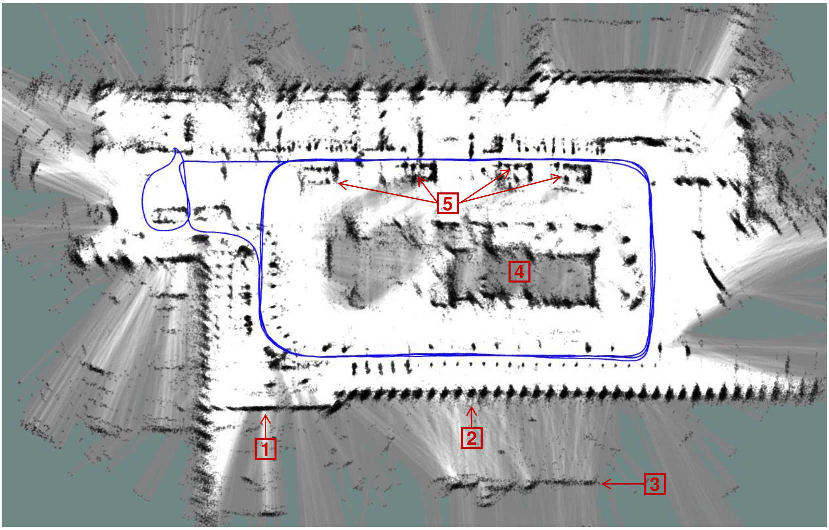
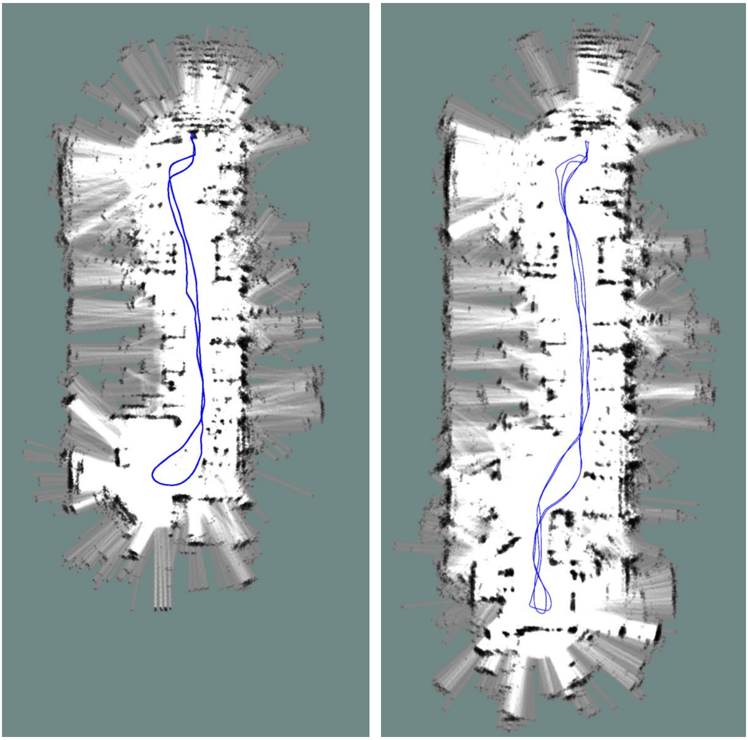


Abbildung 7.6: Radar-SLAM Ergebnis eines getragenen Datensatzes in urbaner Umgebung. Es wurde drei Runden der approximativ gleichen Trajektorie absolviert. Der Schleifenschluss war erfolgreich, ein Drift des Positionsfehlers ist nicht erkennbar.

geflogenen Datensätzen ähnliche Ergebnisse wie bei den zuvor getragenen Datensätzen. Daher kann für geflogene Datensätze die gleiche Performance wie bei den getragenen erwartet werden.

Vergleich zu Laser-SLAM

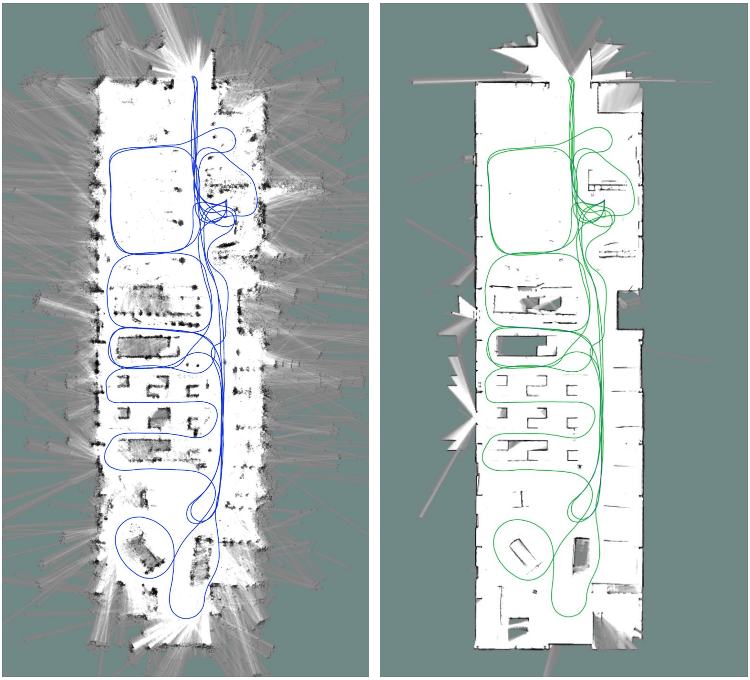
Zur abschließenden Evaluation erfolgt ein qualitativer sowie quantitativer Vergleich in einer realistischen Umgebung zu einem etablierten Laser-SLAM [55]. Hierfür wurde die Multisensorplattform aus Abschnitt 3.1 um einen 2D-Laserscanner erweitert und der Datensatz *DRZ-SLAM-Vergleich* am Deutschen-Rettungsrobotik-Zentrum aufgenommen. Dieser umfasst eine Trajektorienlänge von 609 m über einen Zeitraum von 8,5 min. Das Laser-SLAM soll als Referenzsystem fungieren. Deshalb sind die Daten bei guten Sichtbedingungen und nicht bei beeinträchtigenden Bedingungen, wie Nebel oder Rauch, aufgezeichnet worden. Dabei wird das Ergebnis des Laser-SLAMs als Refe-



(a) Manueller Flug 1 **(b)** Manueller Flug 2

Abbildung 7.7: Radar-SLAM Ergebnis für zwei manuelle Flüge im Deutschen Rettungsrobotik-Zentrum. Der Schleifenschluss hat bei beiden Flügen funktioniert und die Struktur innerhalb der Halle ist gut erkennbar. Sogar der kürzere Flug 1 ermöglicht die Kartierung. Beide Karten weise die gleiche Skalierung auf.

renz betrachtet, um den Radar-SLAM zu evaluieren. Abbildung 7.8 zeigt das Ergebnis des Radar-SLAMs auf der linken Seite sowie das Ergebnis des Laser-SLAMs [55] auf der rechten Seite. Es ist zu erkennen, dass der Laser-SLAM eine sehr genaue Repräsentation der Umgebung liefert. Beispielsweise können Wände klar als gerade Strukturen identifiziert werden. Demgegenüber sinkt die Kartierungsqualität beim Radar-SLAM aufgrund der niedrigeren Auflösung der Rasterkarte und vermehrten Mehrwegeausbreitung. Dennoch lässt die Umgebung sehr gut interpretieren. Der größte Unterschied zeigt sich bei



(a) Radar-SLAM **(b)** Laser-SLAM

Abbildung 7.8: Ergebnis des *DRZ-SLAM-Vergleich*-Datensatzes, der einer Trajektorienlänge von 609 m mit einer Dauer von 8,5 min aufweist. Die Rasterkarte des Radar-SLAMs (links) erreicht fast den gleichen Detailgrad wie die Laser-SLAM Karte (rechts). Wird die Laser-SLAM-Trajektorie als Referenz angesehen, weist die Radar-SLAM-Trajektorie einen ATE von 0,11 m auf.

der Kartierung der Außenwände, da diese den Laserscan nur sehr vereinzelt reflektieren und daher die Wand sehr glatt darstellen. Der Radar-SLAM zeigt hier auf Grund von Reflexionen eine weniger klare Darstellung der Wand. Der qualitative Vergleich beider Karten in Bezug auf die Hindernisse und Strukturen innerhalb der Halle zeigt auch eine hohe Güte. So werden die Strukturen im mittleren Teil der Halle auch durch den Radar-SLAM korrekt kartiert. Die SLAM-Trajektorien in Abbildung 7.8 zeigen eine sehr ähnlichen Verlauf. Unter der Annahme, dass der Laser-SLAM die Referenz darstellt, kann auch eine quantitative Evaluation des Translationsfehlers druchgeführt werden. Dafür wurden beide Trajektorien, wie auch bei der Evaluation der Radar-inertialen

Navigation in Abschnitt 5.4.5, mit Hilfe von *pos-yaw* zueinander ausgerichtet und anschließend der mittlere Positionsfehler (ATE) berechnet. Dabei ergibt sich ein ATE von 0,11 m. Dies ist deutlich kleiner als die Fehler der Radar-inertialen Navigation, siehe Abschnitt 5.4.5, was die hohe Genauigkeit des Radar-SLAMs verdeutlicht. Im Gegensatz zum Laser-SLAM, kann der Radar-SLAM auch bei schlechten Sichtbedingungen etwa durch Nebel, wie am Ende des Kapitels demonstriert wird, bei gleich bleibender Genauigkeit eingesetzt werden. Radar-SLAM liefert somit eine sehr akkurate Positionsschätzung.

7.3 Detektion und Klassifizierung von Personen

Dieser Abschnitt beschreibt die Detektion und Klassifizierung von Personen. Er baut auf den von mir betreuten studentischen Arbeiten [188, 192] auf und ist in weiten Teilen unveröffentlicht. So wurde nur das verbesserte CNN zur Personendetektion in der gemeinsamen Veröffentlichung [184] präsentiert. Ziel ist es Personen, in Thermalbildern zu detektieren und die Körperhaltung hinsichtlich der Klassen *aufrecht*, *sitzend*, *liegend* oder *unbekannt* zu klassifizieren. Dies wird mit Hilfe von Methoden des maschinellen Lernens erzielt. Anstelle eines einzelnen Netzes für die Detektion und Klassifizierung, werden mehrere optimierte und etablierte Netze in einer neuartigen Form kombiniert. Mit diesem entwickelten Ansatz wird eine bessere Generalisierung und auf Grund der effizienten Netz-Architekturen mit *Edge-TPU-Stick* Echtzeitfähigkeit erreicht.

7.3.1 Entwickelte Pipeline

Abbildung 7.9 veranschaulicht die drei Blöcke der entworfenen Pipeline. Zunächst wird eine Vorverarbeitung der Thermalbilder vorgenommen, um Rotationen und Linsenverzerrungen zu entfernen. Anschließend erfolgt die Posenschätzung, was die Detektion der einzelnen Körperteile in Form eines Skeletts im Bild beschreibt. Abschließend wird auf Basis der Posenschätzung die Klassifizierung der Körperhaltung in die Klassen *aufrecht*, *sitzend*, *liegend* oder *unbekannt*. Auf alle drei Blöcke wird im Folgenden näher eingegangen.

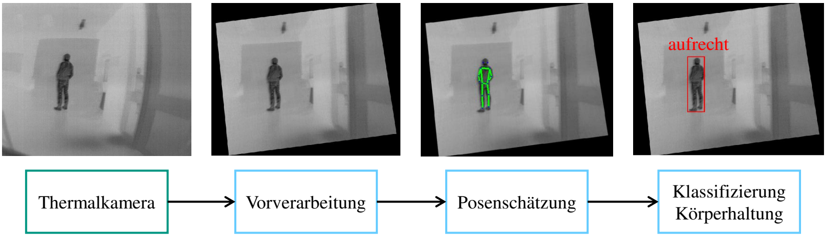


Abbildung 7.9: Veranschaulichung der Detektion und Klassifizierung von Personen

Vorverarbeitung

Ziel des entwickelten Ansatzes ist die Echtzeitfähigkeit auf einem Embedded-Computer zu erreichen. Daher müssen hoch-optimierte CNNs für die Detektion sowie der Bestimmung der Pose von Personen eingesetzt werden. Die Analyse der verfügbaren CNNs hat ergeben, dass die Posenschätzung eine verschlechterte Performance bei Personen liefert, die durch die Linse verzerrt werden oder nicht aufrecht im Bild erscheinen. Dies tritt bei der eingesetzten Thermalkamera und zum Beispiel bei Quadrokopterflügen bei Rollbewegungen auf. Um trotzdem die bestmögliche Performance zu erreichen, wird daher eine Vorverarbeitung der Kamerabilder durchgeführt.

Daher werden Verzerrungen durch die Weitwinkel-Linse der Thermalkamera korriert. Wie in Abbildung 7.9 in den beiden Bildern ganz links zu erkennen ist, erscheinen gerade Linien im Bild nach dem Entfernen von Linsenverzerrungen als gerade. Anschließend wird zur weiteren Verbesserung der Detektionsgüte das Kamerabild um den Rollwinkel korrigiert. Hierzu wird die aktuelle Navigationslösung eingesetzt und das Bild mit Hilfe einer Homographiematrix $\mathbf{H}_{\text{stab}}$ transformiert. Die Berechnung von $\mathbf{H}_{H,\text{stab}}$ erfolgt durch

$$\mathbf{H}_{H,\text{stab}} = \mathbf{K}\,\mathbf{C}_c^{b,T}\,\mathbf{C}_b^s\,\mathbf{C}_c^b\,\mathbf{K}^{-1} \tag{7.1}$$

mit den intrinschen Kameraparameter $\mathbf{K}$, der Rotation vom Kamera- ins körperfeste-Koordinatensystem $\mathbf{C}_c^b$, sowie der Rotation vom körperfesten ins stabilisierte Koordinatensystem $\mathbf{C}_b^s = C(\theta_{sb}, 0, 0)$. Die Transformation des entzerrten Bildes mit $\mathbf{H}_{H,\text{stab}}$ ergibt das Ergebnis der Vorverarbeitung, wie in Abbildung 7.9 im zweiten Bild von links gezeigt ist.

Personendetektion

Die Personendetektion wird mit CNNs durchgeführt. Diese hat die Vorteil, dass neben Personen auch andere Gegenstände detektiert werden können. Zur Echtzeitanwendung werden hier nur Netze betrachtet, die auch mit dem *Edge-TPU-Stick* verwendet werden können. Daher werden nur die durch *Google-Coral* zur Verfügung gestellten Netze [26] evaluiert: *MobileNetV2* [125], *MobileDet* [157] und *EfficentDet* [141]. Wie im Ergebnisteil dargelegt, weist *MobileDet* die höchste Genauigkeit auf.

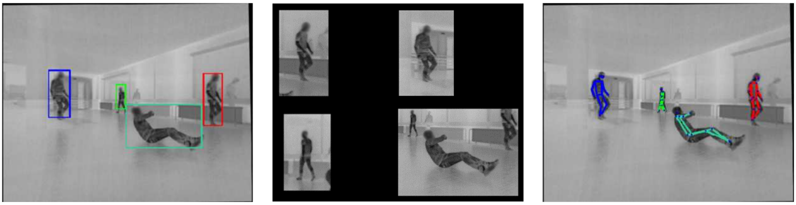
Um die Detektionsperformance noch weiter zu verbessern, wurde in der von mir betreuten Bachelorarbeit [192] ein Verfahren entwickelt, dass effektiv ein sogenanntes Fine-Tuning bestehender CNNs anwendend, was in der gemeinsamen Publikation [184] veröffentlicht wurde. Fine-Tuning ermöglicht es, ein bestehendes CNN mit weiteren Trainingsdaten gezielt auf eine Anwendung hin zu optimieren. Die praktische Umsetzbarkeit hängt hier sehr stark von der Anzahl und der Güte der neuen Trainingsdaten ab. Um dies effektiver zu gestalten, wurde ein neues Verfahren entwickelt [184]. Grundlage hierfür ist, dass die Detektion von Personen in visuellen Bilder eine deutlich höhere Güte aufweist, als bei Thermalbilder. Dies ist insbesondere bei Verwendung von sehr genauen CNNs, die nur auf Grafikkarten ausgeführt werden können, im Vergleich zu den *Edge-TPU-Stick*-Netzen sehr deutlich zu beobachten. Daher wurden weitere Trainingsdaten mit der Multisensorplattform aus Abschnitt 3.1 aufgezeichnet und mit einem sehr genauen CNN (Faster-RCNN [115]) annotiert. Anschließend wurden die Annotationen manuell überprüft und die erstellten Bounding-Boxen mit Hilfe der extrinisischen und intrinsischen Kalibrierung der Kameras aus dem visuellen Bild ins Thermalbild transformiert. So kann effizient ein großer Trainingsdatensatz erstellt werden, ohne eine Vielzahl an Bilder manuell annotieren zu müssen. Für das Fine-Tuning wurden hierbei insgesamt 7597 Bilder aus verschiedenen Umgebungen in Innenräumen und Außenbereichen genutzt, um das verfügbare CNN MobileDet [26] zu optimieren [184].

Die vorverarbeiteten Bilder werden in der Gesamtpipeline aus Abbildung 7.1 mit dem optimierten *MobileDet*-Netz, das im Folgenden *MobileDet-Tuned* genannt wird, verarbeitet, um Personen zu detektieren. Jede Detektion besteht dabei aus der umschließenden rechteckigen Bounding-Box sowie der geschätzten Unsicherheit der Detektion. Hierauf baut die Posenschätzung auf.

Posenschätzung

Ziel ist es Personendetektionen in Form von Bounding-Boxes hinsichtlich ihrer Körperhaltung zu klassifizieren. Um dies zu erreichen, ist es notwending die Position der einzelnen Körperteile und Gelenke zu kennen wie beispielsweise auch in [50, 54] angewendet. Dies wird durch die Posenschätzung erreicht. Hierbei wird das weitverbreitete COCO-Format [83] eingesetzt. Dieses umfasst 17 Merkmalspunkte von Körperteilen wie zum Beispiel der Fuß- Knie, Hüft- und Ellenbogengelenke. Auf Basis dessen lässt sich Körperhaltung sehr gut beschreiben. Fokus bei der Posenschätzung liegt wiederum auf der Echtzeitfähigkeit der Gesamtpipeline auf Embedded-Computern. Daher werden nur neuronale Netze betrachtet, die auf dem *Edge-TPU-Stick* eingesetzt werden können. Von *Google-Coral* sind zwei mögliche Netztypen veröffentlicht worden [27]: *posenet* [106] und *MoveNet* [28]. Beide Netze sind vielversprechend hinsichtlich Genauigkeit und Geschwindigkeit. Während *PoseNet* die Pose von mehreren Personen in einem Bild detektieren kann, kann das verfügbare *MoveNet* nur eine einzelne Pose pro Bild schätzen. Um beide Netze jeweils optimal zu nutzen, sind zwei Ansätze zur Posenschätzung entworfen worden.

Die PoseNet-Pipeline basiert auf der von mir betreuten Masterarbeit [188]. Evaluationen haben gezeigt, dass *PoseNet* Schwierigkeiten hat, weit entfernte Personen im Bild zu detektieren. *PoseNet* wurde auf dem COCO-Datensatz [83] trainiert, der primär im Bild groß erscheinende Personen aufweist, was dieses Verhalten erklärt. Die Klassifizierung von Person ist jedoch auch auf größeren Distanzen von bis zu 10 m notwendig, um den Anwendungsfall der Personenkartierung zu realisieren. Daher wurde in der studentischen Arbeit [188] das folgende Verfahren entwickelt. Auf Basis der Bounding-Boxen der Personendetektion wird ein neues Bild, das sogenannte Region-of-Interest (ROI)-Bild zusammengesetzt. Um Fehler der Personendetektion zu kompensieren, werden die Bounding-Boxen in alle Richtungen um einen Schwellwert erweitert. Das ROI-Bild besteht also nur noch aus den detektierten Personen, wodurch diese deutlich größer im Bild erscheinen. Kleine Bounding-Boxen werden dabei größer und große Bounding-Boxen kleiner skaliert, sodass eine homogene Größe der Personen vorliegt. Die Posenschätzung mit *PoseNet* wird dann auf dem ROI-Bild ausgeführt und die detektierten Posen in das ursprüngliche Bild zurück transformiert. Abbildung 7.10 zeigt das Ergebnis der Personendetektion im linken Bild, das erstellte ROI-Bild in der Mitte und schließlich die zurück transformierte Posenschätzung rechts.



(a) Personendetektion **(b)** ROI-Bild **(c)** Transformierte Posenschätzung

Abbildung 7.10: *PoseNet*-Pipeline mit Erstellung des ROI-Bildes

Die *MoveNet*-Pipeline ist anders aufgebaut. Die verfügbaren *MoveNet*-Modelle können jeweils nur eine einzige Person pro Bild detektieren. Dafür ist *MoveNet* im Vergleich zu *PoseNet* deutlich schneller. Anstatt ein ROI-Bild aufzubauen, wird daher eine Posenschätzung mit *MoveNet* auf jeder Bounding-Box separat ausgeführt. Dies erzeugt eine Detektionslaufzeit, die proportional zur Anzahl an Personendetektionen ist. In den betrachteten Szenarien sind jedoch immer nur maximal fünf Personen in einem einzelnen Bild abgebildet. Daher werden in der entworfenen MoveNet-Pipeline maximal fünf Posenschätzungen pro Bild ausgeführt.

Das Ergebnis beider Pipelines ist die Posenschätzung, also der 17 Merkmalspunkte sowie der geschätzten Unsicherheit der Detektion. Basierend darauf wird die Klassifizierung ausgeführt. Wurde für eine Person zwar eine Bounding-Box jedoch keine Pose detektiert, wird Körperhaltung als *unbekannt* klassifiziert.

Klassifizierung der Körperhaltung

Die Klassifizierung der Körperhaltung führt auf Basis der Posenschätzung eine Einteilung in die Klassen *aufrecht*, *sitzend* oder *liegend* ein. Dies kann auf Basis eines Merkmalsvektors mit gängigen Verfahren des überwachten maschinellen Lernens mittels eines Datensatzes umgesetzt werden [48]. Typischerweise werden dafür Support-Vector-Machines (SVMs), logistische Regression (LR), Enscheidungsbäume (EBs) oder Radom-Forests (RFs) eingesetzt. Der Merkmalsvektor wird mit Hilfe der Posenschätzung aufgebaut. Basierend auf der von mir betreuten studentischen Arbeit [188] weist der entworfene Merkmals-

vektor die folgenden Komponenten auf: die absolute Orientierung des Oberkörpers, den Winkel des Hüftgelenks, das Verhältnis der Höhe und Breite der detektierten Bounding-Boxen und die Konfidenz der Posenschätzung Somit wird aus dem Ergebnis der Posenschätzung der Merkmalsvektor aufgestellt. Die Position der Arme und Knie wird nicht berücksichtigt, da diese ähnliche Posen sowohl bei *aufrechten*, *sitzenden* als auch *liegenden* Personen aufweisen können.

Für das Training wurden selbst erstellte Datensätzen mit der Multisensorplattform aufgezeichnet, die jeweils eine einzige Person in allen drei Posen umfangreich abbildet. Insgesamt liegen so circa 12 500 Bilder aller drei Posen vor. Diese wurde mit der zuvor beschriebenen Posenschätzung prozessiert und alle Bilder, bei denen diese erfolgreich war, dem Datensatz hinzugefügt. Da die Posenschätzung, wie auch im Ergebnisteil evaluiert, nicht bei allen Thermalbildern erfolgreich ist, enthält der Datensatz weniger Detektionen als Bilder vorhanden sind. So war die Posenschätzung auf circa 10 500 Bildern erfolgreich, die somit den Datensatz bilden.

Alle vier genannten Algorithmen (SVM, LR, EB, RF) wurde schließlich mit dem Framework *SciPy* [148] implementiert und trainiert. Dabei wurden 75 % des Datensatzes als Trainingsdatensatz und die restlichen 25 % als Validierungsdatensatz eingeteilt. Die Auswahl der Samples erfolgte dabei zufallsbasiert. Das Training wurde dann mit dem Trainingsdatensatz ausgeführt. Mittels des Validierungsdatensatz, können alle vier Ansätze evaluiert werden, wobei der Random-Forest-Ansatz die besten Ergebnisse erzielt hat. Der trainierte Random-Forest ist wie in Tabelle 7.1 hinsichtlich der Metriken Precision und Recall, siehe Abschnitt 2.5 im Grundlagenkapitel, evaluiert. Es zeigt sich, dass der Random-Forest eine sehr hohe Güte erzielt, so erreicht die Klassifizierung auf den Validierungsdatensatz für die Precision und Recall, siehe Abschnitt 2.5 im Grundlagenkapitel, durchschnittlich 98 %.

Metrik	aufrecht	sitzend	liegend
Precision	98 %	98 %	97 %
Recall	100 %	97 %	97 %

Tabelle 7.1: Trainingsergebnis des Random-Forests für Precision und Recall auf dem Validierungsdatensatz bestehen aus 2635 Bildern.

7.3.2 Ergebnisse

In diesem Abschnitt werden zunächst die neuronalen Netze für Personendetektion und Posenschätzung an sich und anschließend die Gesamtpipeline evaluiert. Diese werden mit umfangreichen, realistischen Testdatensätzen der Multisensorplattform hinsichtlich Genauigkeit und Rechenzeit evaluiert. Die Testdatensätzen bestehen aus den beiden Datensätzen *Sinlge-Person*, mit einer Person, und *Multi-Person*, mit bis zu vier Personen im gleichen Thermalbild, und umfassen insgesamt 2748 Bilder in unterschiedlichen Szenen. Dabei nehmen die Personen die in dieser Arbeit betrachteten Körperhaltungen *aufrecht*, *sitzend* oder *liegend* ein. Von den 2748 Bildern sind 2362 gelabelt, weisen also eine Referenz in Bezug auf die Bounding-Box der Personen sowie Körperhaltung auf. Da auch mehrere Personen pro Bild auftreten können, ist die Anzahl an annotierten Personen mit 3540 höher als die Anzahl an Bildern. Die Implementierung erfolgte in Python unter der Nutzung des Frameworks *Tensorflow* [3].

Benchmark Objektdetektion

Die Objektdetektion ist Grundbestandteil der Posenschätzung. Da der Fokus auf der Echtzeitfähigkeit mit Embedded-Computern liegt, werden nur Netze betrachtet, die auch auf dem *Edge-TPU-Stick* prozessiert werden können. Als Metrik wird die oft genutzte Metrik Intersection-over-Union (IoU), siehe Abschnitt 2.5 der Grundlagen, eingesetzt. Dabei wird die Bounding-Box, wie üblich [118], als korrekt eingestuft, wenn die IoU größer als 50 % ist.

Tabelle 7.2 zeigt die Ergebnisse für die drei vielversprechendsten Netze, die öffentlich verfügbar sind [26]. Die aufgeführten Detektionsraten sind mit Hilfe des annotierten Datensatzes entsprechend der drei Körperhaltungen aufgeschlüsselt. Es zeigen sich sehr hohe Detektionsraten von 93 – 98 % bei *aufrechten* und *sitzenden* Personen bei allen vier betrachteten Netzen. Große Unterschiede von 60 – 77 % zeigen sich jedoch bei *liegenden* Personen. Hier erreicht das MobileDet-Netz mit 63 % das beste Ergebnis der verfügbaren Netze.

Mit dem in dieser Arbeit neu trainierte Netz *MobileDet-Tuned* konnte hingegen die Detektionrate bei liegenden Personen um 14 % erhöht werden. Über alle

	MobileNetV2 [125]	*EfficientDet* [141]	*MobileDet* [157]	*MobileDet-Tuned*
aufrecht	96 %	96 %	97 %	96 %
sitzend	94 %	98 %	96 %	99 %
liegend	60 %	62 %	63 %	77 %
Mittelwert	83 %	85 %	85 %	90 %
Laufzeit	23 ms	60 ms	20 ms	20 ms

Tabelle 7.2: Detektionsraten für IoU = 0,5 und Laufzeiten der CNNs zur Personendetektion prozessiert auf dem *Intel-NUC-i7* mit *Edge-TPU-Stick*.

Klassen wird eine Verbesserung um 5 % erreicht, ohne dass sich die Laufzeit erhöht. So werden zur Objektdetektion im Schnitt lediglich 20 ms auf dem *Inte-NUC-i7* mit *Edge-TPU-Stick* benötigt. Auf Basis dieser Ergebnisse wird das *MobileDet-Tuned*-Netz zur Personendetektion im Folgenden eingesetzt.

Benchmark Posenschätzung

Zur Evaluation der Posenschätzung werden die Datensätze mit der *PoseNet*-Pipeline und der *MoveNet*-Pipeline prozessiert. Eine umfangreiche Evaluation aller verfügbaren *PoseNet*-Modelle wurde in der von mir betreuten studentischen Arbeit [188] durchgeführt. Die zwei vielversprechendsten *PoseNet*-Modelle sind in Tabelle 7.3 dargestellt. Für *MoveNet* liegen nur zwei Netz-Modelle vor: *Lightning* und *thunder*. Zum Vergleich werden die Datensätze auch mit einem der genausten Netze zur Posenschätzung auf einem Dektop-Computer mit Grafikkarte verarbeitet. Hierbei wird *mmPose* [25] mit dem *ResNet*-152-Backbone verwendet. Da die Detektionsraten bei *mmPose* auf dem ROI-Bild schlechter waren, wird dieses bei *mmPose* nicht verwendet.

Tabelle 7.3 fasst die Ergebnisse der Posenschätzung zusammen und schlüsselt die Personen anhand der Körperhaltung auf. Für jede Klasse wird der Anteil der erfolgreichen Posenschätzungen ermittelt. Um eine Detektion als erfolgreich einzustufen zu können, müssen alle Merkmale der Posenschätzung innerhalb der annotierten Bounding-Box liegen. Da dies als Bedingung für die anschließende Klassifikation ausreicht, wird auf eine merkmalsspezifische Untersuchung verzichtet.

Hardware	*Intel-NUC-i7* mit *Edge-TPU-Stick*				*Desktop PC mit GPU
Netz	*PoseNet* [106]		*MoveNet* Single Pose [28]		*mmPose* [25]
Backbone	ResNet-50	ResNet-50	Lightning	Thunder	ResNet-152
Eingangsgröße	768 × 496	928 × 672	192 × 192	256 × 256	512 × 512
Aufrecht	97 %	97 %	94 %	95 %	94 %
Sitzend	94 %	95 %	94 %	94 %	96 %
Liegend	27 %	37 %	57 %	58 %	68 %
Mittelwert	73 %	76 %	81 %	82 %	86 %
Laufzeit pro Bild	121 ms	840 ms	22 ms	51 ms	680 ms*

Tabelle 7.3: Detektionsraten und Laufzeiten der besten *PoseNet*-Netze auf dem *Inte-NUC-i7* mit *Edge-TPU-Stick*.

Wie auch bei der Personendetektion zeigen sich bei der Posenschätzung sehr gute Detektionsraten von über 90 % für *aufrechte* und *sitzende* Personen. *Liegende* Personen werden dagegen von allen Netzen deutlich schlechter erkannt, da die notwendigen Merkmale deutlich seltener als erfolgreich eingestuft werden. *PoseNet* erreicht hier mit 27 % und 37 % die schlechtesten Ergebnisse. Mit *MoveNet* lässt sich die Detektionsrate für die Klasse *liegend* deutlich steigern. *MoveNet* ist im Gegensatz zu *PoseNet* neben dem *COCO*-Datensatz [83] mit einem weiteren Google-internen Datensatz trainiert worden [28]. Dieser weist eine Vielzahl verschiedener Personenposen im Alltag aber auch bei Sport auf und ermöglicht daher eine bessere Detektion von zum Beispiel *liegenden* Personen. Ein Fine-Tuning der Posenschätzung speziell mit liegenden Personen und Thermalbildern kann hier nicht eingesetzt werden, da die benötigten Information für *PoseNet* und *MoveNet* im Gegensatz zum *MobileDet* nicht veröffentlicht sind. Da *PoseNet* und *MoveNet* bei der implementierten Pipeline auf die Bounding-Boxes angewiesen sind, sind die Detektionsraten durch die Personendetektion von *MobileDet-Tunded* limitiert.

Das beste Ergebnis bei der Poseschätzung wird durch *mmPose* erreicht. So erreicht *mmPose* im eine Detektionsrate von durchschnittlich 86 %. Die *MoveNet*-Pipeline erzielt jedoch mit 81 % und 82 % nur geringfügig schlechtere Ergebnisse. Im Gegensatz zu *mmPose* erzielt die *MoveNet*-Pipeline viel geringere Laufzeiten sogar auf einem Embedded-Computer mit *Edge-TPU-Stick*. Die *MoveNet*-Pipeline erzielt auch gegenüber der *PoseNet*-Pipeline deutlich bessere Laufzeiten.

Gesamtpipeline

Die Evaluation der Gesamtpipeline erfolgt analog zur Evaluation der Posenschätzung. Dabei wird die Erfolgsrate betrachtet, also wie viele der Personen detektiert und hinsichtlich der Körperhaltung korrekt klassifiziert wurden. Die Ergebnisse sind in Tabelle 7.4 aufgeführt.

Die *PoseNet*-Pipeline erreicht mit dem ROI-Bild über alle Klassen eine deutliche Verbesserung. So ist die *PoseNet*-Pipeline mit ROI-Bild im Schnitt um 8 % besser. Da *PoseNet* liegende Personen am schlechtesten erkennt, erreicht die Gesamtpipeline für diese Klasse die schlechtesten Ergebnisse. Die *MoveNet*-Pipeline erzielt mit einem Mittelwert von 78 % und 79 % deutlich bessere Ergebnisse. Dabei ist die Verbesserung vor allem bei *liegenden* Personen zu erkennen. Die *mmPose*-Pipeline erreicht mit 85 % die besten durchschnittlichen Erfolgsraten. Dies ist aber lediglich mit einem Desktop-Computer mit Grafikkarte realisierbar.

Das Ergebnis der Gesamtpipeline spiegelt das Ergebnis der Posenschätzung wider. Die Klassifizierung der Körperhaltung mit dem Random-Forest kann mit sehr hoher Güte durchgeführt werden. So werden über alle Klassen hinweg bezüglich Precision und Recall durchschnittlich 98 % erreicht.

Ziel ist die Detektion und Klassifizierung von Personen in Echtzeit mit dem *Intel-NUC-i7* und dem *Edge-TPU-Stick*. Um hierfür eine Rate von mindestens

Hardware	*Intel-NUC-i7* mit *Edge-TPU-Stick*				*Desktop PC mit GPU
Netz	*PoseNet* 768 × 496		MoveNet Single Pose		*mmPose*
Konfiguration	ohne ROI	mit ROI	Lightning	Thunder	ohne ROI
Aufrecht	90 %	95 %	90 %	91 %	92 %
Sitzend	77 %	89 %	91 %	90 %	94 %
Liegend	20 %	26 %	54 %	57 %	66 %
Mittelwert	62 %	70 %	78 %	79 %	85 %
Laufzeiten pro Bild	166 ms	198 ms	65 ms	99 ms	720 ms*

Tabelle 7.4: Evaluation der Gesamtpipeline und Vergleich zur Pipeline mit Desktop Computer und Grafikkarte (GPU). Die *Edge-TPU-Stick*-Pipeline erzielt im Schnitt 15 % schlechtere Detektionsraten als die GPU-Pipeline benötigt aber nur etwa ein Drittel der Laufzeit und kann auf einem Embedded-Computer eingesetzt werden.

10 Hz zu gewährleisten, wird im Folgenden die *MoveNet*-Pipeline in der Konfiguration *lightning* eingesetzt. Diese weist eine durchschnittliche Laufzeit von 65 ms auf und erreicht daher die geforderte Rate. Im Gegensatz zur *thunder*-Konfiguration erzielt die *lightning*-Konfiguration 34 % schnellere Laufzeiten bei nur 2 % durchschnittlicher Einbuße der Erfolgsrate. Abbildung 7.11 zeigt beispielhaft Ergebnisse dieser Pipeline aus den beiden Datensätzen *Single-Person* (oben) und *Multi-Person* (unten). Unterschiedliche Perspektiven und Entfernungen zur Thermalkamera sowie Körperhaltungen demonstrieren die Leistung der ausgewählten Pipeline. Bei liegenden Personen wird mit dieser Pipeline nur eine Erfolgsrate von 45 % erreicht. Dennoch liegt die Detektionsrate der Personendetektion mit dem Netz *MobileDet-Tuned*, siehe Tabelle 7.2, bei 77 % für *liegende* Personen. Es werden also insgesamt 77 % der *liegenden* Personen detektiert aber nur 45 % als liegend klassifiziert, weil die Posenschätzung häufig fehlschlägt. Dennoch sind diese Personen mit der Klasse *unbekannt* versehen und werden trotzdem kartiert. Die Kartierung wird im nächsten Kapitel beschrieben.

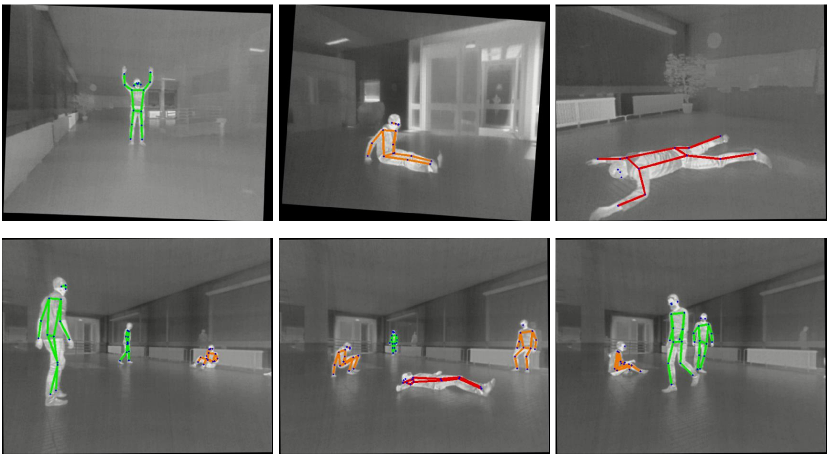


Abbildung 7.11: Beispielhafte Ergebnisse der beiden Testdatensätze *Single-Person* (oben) und *Multi-Person* (unten). Die klassifizierten Zustände sind farblich markiert: *Aufrecht* (grün), *sitzend* (orange) und *liegend* (rot).

7.4 Personenkartierung

Die zuvor beschriebenen Aspekte werden nun in der Personenkartierung zusammengefasst. Diese basiert auf den Radarscans, der SLAM-Karte und -Trajektorie sowie der Personendetektion mit Klassifizierung wie in Abbildung 7.1 gezeigt ist. Um die Position der im Bild detektierten Personen im metrischen Raum zu schätzen, wird in dieser Arbeit ein Kalman-Filter eingesetzt. Da die SLAM-Karte aufgrund der Schleifenschlussfähigkeit über mehrere Beobachtungen korrigiert werden kann, erfolgt die Positionsschätzung bezüglich eines lokalen Koordinatensystems, das zum Zeitpunkt der ersten Beobachtung festgelegt wird. Wird die SLAM-Trajektorie nachträglich korrigiert, wird damit die Position der kartierten Person ebenso korrigiert, da die Relation zum Kartenkoordinatensystem zu einem festgelegten Zeitpunkt bekannt ist. Jede Person wird mit Hilfe eines Kalman-Filters mit dem folgenden Zustand $\boldsymbol{x}$ zum Zeitpunkt k modelliert

$$\boldsymbol{x}_k = \left[p_x^{\mathrm{b}}, p_y^{\mathrm{b}}\right]^T, \tag{7.2}$$

wobei die Position im körperfesten Koordinatensystem, der Anker-Pose, geschätzt wird. Wenn eine neue Messung vorliegt, werden die Zustände aller sich im Sichtbereich befindlichen Personen propagiert. Hierdurch steigt entsprechend des darunterliegenden Systemmodells die Unsicherheit der Zustände an. Kann die neue Beobachtung zu einem propagierten Zustand assoziiert werden, wird der entsprechende Zustand im Filterupdate korrigiert und dessen Unsicherheit verkleinert. Für alle anderen Zustände wächst mit jedem Ausbleiben einer Beobachtung die Unsicherheit weiter. Die Propagation wird mit einem isotropischen Rauschen $\mathbf{Q}_k$ ausgeführt:

$$\mathbf{P}_{k+1} = \mathbf{P}_k + \mathbf{Q}_k. \tag{7.3}$$

Das Filterupdate wird in mehreren Schritten für jede Personen-Messung angewendet. Zunächst wird die Personen-Messung im Koordinatensystem der SLAM-Karte aufgestellt. Hierfür werden die jeweils aktuellen Radarscans der Radarsensoren sowie die Bounding-Box der Personendetektion im Bild der Thermalkamera betrachtet. Die Bounding-Box liefert zwar die Richtung der Detektion, da aber keine genaue Distanz-Information im Bild vorliegt, muss diese über die Radarscans geschätzt werden. Dies erfolgt durch die Projekti-

on der Radarscan-Punkte in das Bild der Thermalkamera. Dafür werden alle N-Scan-Punkte $\boldsymbol{p}^r_{i,l}, i \in N$ mit Hilfe der extrinsichen Kalibierung des l-ten Radarsensors ($\boldsymbol{l}^b_{br,l}, \mathbf{C}^b_{r,l}$) in das körperfeste Koordinatensystem transformiert:

$$\boldsymbol{p}^b_i = \mathbf{C}^b_{r,l}\boldsymbol{p}^r_{i,l} - \boldsymbol{l}^b_{br,l}. \tag{7.4}$$

Anschließend erfolgt die Projektion in Bild-Koordinaten unter Verwendung der extrinsischen und intrinsischen Kameraparameter. Befindet sich mindestens ein Radarscan-Punkt innerhalb der Bounding-Box, kann die Position der Person im körperfesten Koordinatensystem $\boldsymbol{p}^b_{\text{Person}}$ mit Hilfe von (7.4) aufgestellt werden. Liegen mehrere Punkte innerhalb der Bounding-Box, wird der Scan-Punkt mit dem besten Signal-zu-Rauschverhältnis ausgewählt. Bei Personen, die mehr als circa 7 m entfernt sind, liegen häufig keine Radarscan-Punkte mehr vor, da diese auf Grund des geringen RCS mit den eingesetzten Radarsensoren nicht mehr detektiert werden. Um dennoch ein Filterupdate ausführen zu können, wird die Distanz zur Person approximiert. Laut [79] ist die durchschnittliche Größe von Männer und Frauen in Deutschland 1,73 m. Wurde die Personen als *aufrecht* klassifiziert, kann somit die Distanz zur Person mit Hilfe der Merkmalspunkte aus der Posenschätzung approximiert werden. Analog kann dies auch für sitzende Personen basierend auf der durchschnittlichen Torsolänge durchgeführt werden. Ist die Person als *liegend* oder *unbekannt* klassifiziert, wird dieses Verfahren nicht angewendet.

Mit Hilfe der aktuellen Position aus der SLAM-Trajektorie erfolgt die Transformation von $\boldsymbol{p}^b_{\text{Person}}$ ins Karten-Koordinatensystem:

$$\boldsymbol{p}^k_{\text{Person,Messung}} = \mathbf{C}^k_b\boldsymbol{p}^b_{person} + \boldsymbol{p}^k. \tag{7.5}$$

Die Assoziation stellt einen wichtigen Schritt dar. Die Implementierung erfolgt dabei Distanz-basiert. Dazu werden alle bereits kartierten Personen mit Hilfe der korrespondierenden Anker-Posen im SLAM-Graph ins Karten-Koordinatensystem überführt:

$$\boldsymbol{p}^k_{\text{Person,Filter}} = \mathbf{C}^k_{b,\text{Anker}}\boldsymbol{p}^b_{\text{person,Filter}} + \boldsymbol{p}^k_{\text{Anker}}. \tag{7.6}$$

Nun kann mit Hilfe der bereits geschätzten Personen-Positionen $\boldsymbol{p}^k_{\text{Person,Filter}}$ und der Messung $\boldsymbol{p}^k_{\text{Person,Messung}}$ die jeweilige Distanz ermittelt werden. Die

Assoziierung ist erfolgreich, wenn die Distanz unterhalb eines Schwellwertes liegt. Ist eine Assoziierung nicht möglich, wird ein neuer Kalman-Filter für die gemessene Personen-Position angelegt. Die initiale Kovarianz $\mathbf{P}_{\text{Init}}$ modelliert dabei die Charakteristik der Messung. In Polarkoordinaten ausgedrückt weist diese eine geringe Unsicherheit des Winkels und eine größere Unsicherheit der Distanz auf. Die Richtung wird durch den Mittelpunkt der Bounding-Box oder, wenn vorhanden, über den Mittelwert aller Merkmale der Posenschätzung gebildet und weist somit eine sehr hohe Genauigkeit auf. Die Entfernung wird entweder über den Radarscan oder die Approximation der Personengröße bestimmt. Während der Radarscan als sehr genau eingestuft wird, wird für die Approximation über die Personengröße eine deutlich größere Unsicherheit gewählt.

Bei erfolgreicher Assoziierung wird ein Kalman-Filter-Update ausgeführt. Dieses ist linear implementiert, da die Messung auch Teil des Filterzustandes ist. Die Rauschmatrix $\mathbf{R}_m$ wird analog zur initialen Kovarianz $\mathbf{P}_{\text{Init}}$ gebildet. Schließlich wird auch eine Schätzung der Körperhaltung mit Hilfe eines gleitenden Fensters durchgeführt. Hierbei werde die Ergebnisse der Klassifikation über einen Zeitraum von fünf Messungen betrachtet. Die Klasse, die dabei am häufigsten ermittelt wurde, wird als Körperhaltung der Person angenommen.

7.4.1 Ergebnisse

Die Demonstration der Gesamtpipeline erfolgt mit Hilfe eines weiteren autonomen Fluges am Deutschen Rettungsrobotik-Zentrum (DRZ). Der Flug wurde mit der Flugplattform aus Abschnitt 3.2 durchgeführt und erfolgte analog zur Demonstration der online Radar-inertialen Navigation aus Abschnitt 5.4.5. Um Echtzeitfähigkeit auf dem *Intel-NUC-i7* zu erzielen und damit eine online Kartierung zu ermöglichen, werden die Thermalbilder mit einer Rate von 10 Hz verarbeitet. Dabei befanden sich insgesamt drei Personen jeweils seitlich des Flugpfads im dichten Nebel. Abbildung 7.12 zeigt den zeitlichen Verlauf des Flugs und der Kartierung. Jede Spalte stellt verschiedenen Visualisierungen des jeweils gleichen Zeitpunktes dar, wobei 3 s, 12 s und 20 s nach dem Start betrachtet werden. Die obere Zeile zeigt die SLAM-Rasterkarte mit SLAM-Trajektorie sowie der Personen-Kartierung. Die zweite Zeile veranschaulicht die Thermalbilder mit Bounding-Boxen der Personendetektionen sowie die Radarscan-Punkte des linken (rot) und rechten (blau) Radarsensors. Die dritte

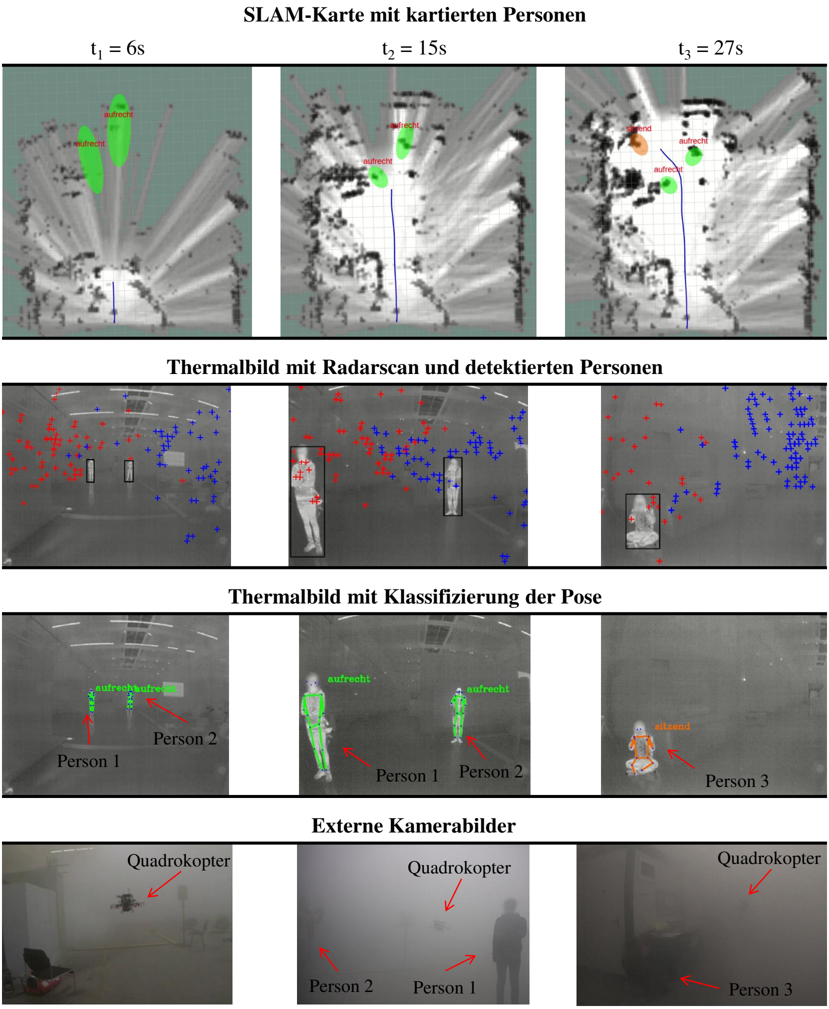


Abbildung 7.12: Zeitlicher Verlauf der online Kartierung für einen autonomen Flug in dichtem Nebel. Die Bilder jeder Spalte gehören dabei zum gleichen Zeitpunkt wobei 3 s, 12 s und 20 s nach Start betrachtet werden. Die Kreise in der Karte markieren die Unsicherheit der Personen-Position und nehmen mit der Anzahl an Messungen ab. Die Kreuze in der zweiten Zeile zeigen den Radarscan des linken (rot) und rechten (blau) Radarsensors. Die Personen können samt der Körperhaltung korrekt kartiert werden.

Zeile stellt die Ergebnisse der Posenschätzung mit dem Klassifikationsergebnis hinsichtlich der Körperhaltung dar. Die unterste Zeile zeigt Bilder des Flugs von externen Kameras aus verschiedenen Perspektiven.

Der schrittweise Aufbau der Karte ist in der oberen Zeile gut zu erkennen und bildet die Umgebung der DRZ-Halle detailliert ab. Der Bereich vor der aktuellen Position der Flugplattform ist insbesondere nach dem Zeitpunkt t_1 sehr detailliert und Hindernisse sind klar erkennbar. Die grünen Ellipsen innerhalb der Karte zeigen den ± 3-σ-Bereich der geschätzten Personenpositionen.

Bei der Visualisierung der Radarscans in der zweiten Zeile zeigen sich eine Vielzahl der Detektionen im Bereich der Decke, da die dort befindlichen Doppel-T-Träger starke Radarreflektoren darstellen. Bereits vor dem Zeitpunkt t_1 können zwei Personen detektiert und klassifiziert werden. Da jedoch noch keine Radarscan-Punkt auf Grund der großen Distanz für die beiden Personen vorlagen, wurde die Distanz über die Größe approximiert und weist daher in Blickrichtung noch eine sehr große Unsicherheit auf. Im Gegensatz dazu ist die Unsicherheit der Blickrichtung viel kleiner, da dieser über die Personendetektion mit hohe Güte geschätzt wird. Gut zu erkennen ist die stark abnehmenden Unsicherheit der Position der Personen 1 und 2 zum Zeitpunkt t_2. Die Radarscans liefern hier eine genaue Messung der Distanz, was durch die adaptive Rauschmatrix im Kalman-Filter-Update modelliert wird. Die Detektion der *sitzenden* Person 3 erfolgt initial nicht korrekt, wird jedoch durch mehrmalige Beobachtungen bis zum Zeitpunkt t_3 korrekt klassifiziert.

Abbildung 7.13 zeigt das finale Ergebnis der Kartierung. Alle drei Personen konnten korrekt bezüglich der Position kartiert sowie der Körperhaltung klassifiziert werden. Rettungswege sind anhand der Rasterkarte gut erkennbar und mit Hilfe des klassifizierten Zustands kann eine Priorisierung des Rettungseinsatzes gezielt erfolgen. Wie die untere Bildzeile in Abbildung 7.12 veranschaulicht, fand der Flug in dichtem Nebel mit reduzierten Sichtweiten unterhalb eines Meters statt. Dennoch kann das entwickelte Sensorkonzept und die entwickelte Kartierungs-Pipeline die Umgebung umfassend während eines autonomen Fluges abbilden und freie Bereiche (in weiß) sowie Hindernisse (in schwarz) klar darstellen. Dadurch wir die Stärke des Sensorkonzepts in Kombination mit dem Navigationsansatz verdeutlicht: Die Navigation kann unabhängig von den Sichtbedingungen erfolgen und die Umgebung sowie Personen können genau kartiert werden.

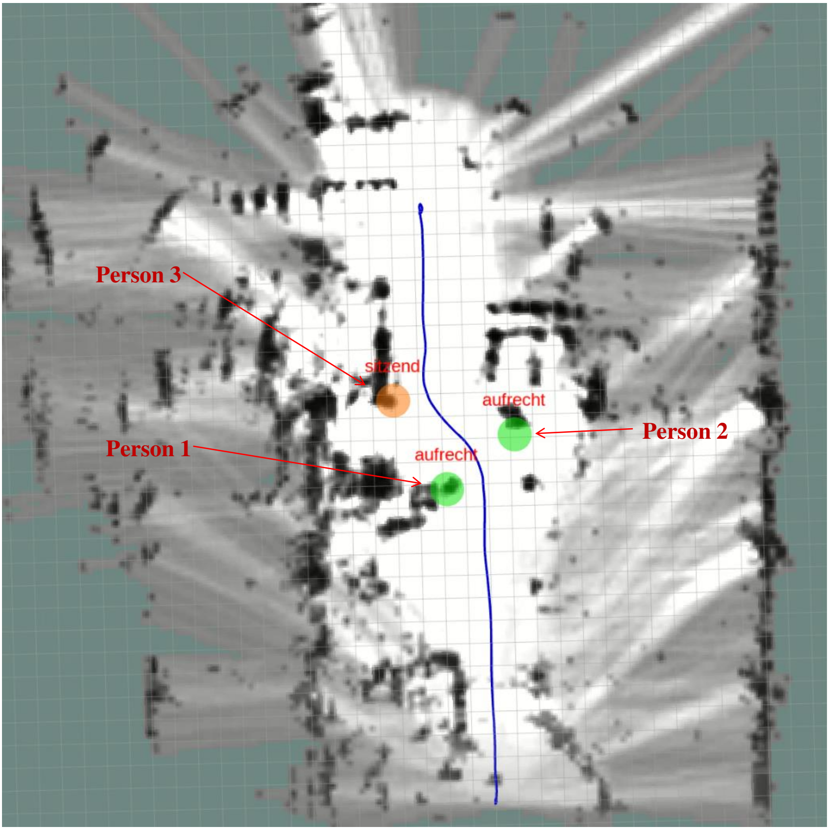


Abbildung 7.13: Detailansicht der erstellten Karte des autonomen Fluges in dichten Nebel. Die Personen konnten erfolgreich kartiert sowie der Zustand geschätzt werden. Mit Hilfe der Rasterkarte sind direkte Rettungswege erkennbar. Aufrechte Personen sind grün markiert und die sitzende Person in orange womit eine Priorisierung des Rettungseinsatzes erfolgen kann.

7.5 Fazit

In diesem Abschnitt erfolgte die Beschreibung der entwickelten Kartierungspipeline. Diese ermöglicht die Kartierung der Umgebung in Form einer Rasterkarte sowie die Detektion und Verortung von Personen unabhängig der Sichtbedingungen. Hierfür wurde auf den kombinierten Inlier-Radarscan so-

wie die Navigationslösung des Radar-inertialen Navigationsfilters zurückgegriffen. Die entwickelte Filterung der Radarscans ermöglicht die Nutzung eines etablierten Laser-SLAM-Verfahrens. So wird die Navigation um Schleifenschlüsse und eine Rasterkartierung erweitert. Dies führt zu einem langzeitstabilen Navigationsansatz. Ein Vergleich zu etablierten Laser-SLAM zeigt den hohen Detailgrad sowie Genauigkeit des Radar-SLAMs. Im Gegensatz zu Laser-SLAM wird dies auch bei schlechten Sichtbedingungen erreicht und liefert damit beispielsweise die Grundlage für lokale Trajektorienplanung zur Kollisionsvermeidung

Insbesondere für Such- und Rettungsmissionen ist neben einer Rasterkarte auch die Position und der Zustand von Personen notwendig. Daher wurde ein neuartiger Ansatz entwickelt, der Personen im Thermalbild detektiert und die Körperhaltung klassifiziert. Dafür wurden moderne neuronale Netze mit dem Ziel kombiniert, neben einer hohen Detektionsrate auch die Echtzeitfähigkeit auf dem Embedded-Computer des Fluggerätes zu ermöglichen. Umfangreiche Evaluationen zeigen die hohe Leistungsfähigkeit der entwickelten Pipeline. So werden nur 5 % schlechtere Detektionsraten bei einer siebenfach schnellere Laufzeiten als mit einer Referenzimplementierung erreicht. Im Gegensatz zur Referenzimplementierung, die einen Desktop-Computer und Grafikkarten nutzt, ist die entwickelte Pipeline mittels des *Edge-TPU-Sticks* auf dem Embedded-Computer des Fluggerätes einsetzbar.

Basierend auf dem Radarscan, Radar-SLAM sowie der Personendetektion und Klassifizierung, erfolgt die Personenkartierung. Dabei wird jede gefundene Person mit einem Kalman-Filter beschrieben und so Messunsicherheiten adäquat modelliert. Die Personen werden relative zur SLAM-Trajektorie verankert, um eine robuste Kartierung auch bei Schleifenschlüssen zu gewährleisten. Die Demonstration eines autonomen Fluges in dichtem Nebel zeigt die Güte der entwickelten Kartierungspipeline. Alle Personen konnten bezüglich der Position und Körperhaltung korrekt kartiert werden. Im Kontext von Such- und Rettungsmissionen wird so ein umfangreiches Lagebild erstellt, das die effektive Rettung ermöglicht.

8 Zusammenfassung

Ziel dieser Arbeit war die Entwicklung von Verfahren zur Navigation autonomer Fluggeräte sowie die Kartierung der Umgebung trotz schlechter Sicht- und GNSS-Bedingungen. Hierzu wurde ein passendes Sensorkonzept entwickelt, dass im Wesentlichen Radarsensoren, eine Thermalkamera und Inertialsensorik kombiniert. Durch die geringe Größe und das niedrige Gewicht eignet sich diese auch für kleine Fluggeräte.

Der Start und die Landung sind kritische Phasen eines Fluges und benötigen eine Navigationsstützung höchster Zuverlässigkeit. Dies ist in schlechten Sicht- und GNSS-Bedingungen mit konventionellen Systemen für kleine Fluggeräte nicht verfügbar. Daher wurden zwei neuartige Verfahren entwickelt. Das rein Radar-basierte System benötigt lediglich den bereits zur Höhenstützung verbauten Radarsensor sowie eine passive Landeplattform mit Radarreflektoren. Das Thermalkamera-basierte System nutzt eine nach unten gerichtete Thermalkamera in Kombination mit einer Landeplattform, in die Thermalmarker integriert sind. Beide Systeme ermöglichen autonome Start- und Landemanöver, wie ausführlich evaluiert und demonstriert wurde, was die Leistungsfähigkeit beider Systeme beweist. Dabei erzielt das System mit Thermalkamera eine bessere Performance hinsichtlich der Genauigkeit, benötigt mit der Thermalkamera jedoch einen zusätzlichen Sensor.

Ein zuverlässiges und akkurates Navigationssystem ist die Grundlage für autonome Fluggeräte. Hierfür wurde in dieser Arbeit ein neuartiges Radar-inertiales Navigationssystem entwickelt, dass die Limitierungen von Kamera- und Laser-basierten Ansätzen hinsichtlich schlechter Sichtbedingungen löst. Basierend auf Radarscans, wurden neue Verfahren zur Schätzungen der Eigengeschwindigkeit und des Gierwinkels selbst bei dynamischen Flugmanövern und in herausfordernden Umgebungen erarbeitet. Die Fusion mit Inertialsensorik resultiert in einem sehr genauen und effizienten neuartigen Navigationssystem. Dieses kann eine frei parametrierbare Anzahl an Radarsensoren fusionieren, wobei auch die extrinsische Kalibrierung der Sensoren online geschätzt

wird. Die entwickelte Pipeline *x-RIO* wurde mit umfangreichen Datensätzen in Innenräumen sowie in urbanen Umgebungen evaluiert. Dabei wurden sogar in guten Sichtbedingungen bessere Ergebnisse als mit etablierten Kamera-basierten Ansätzen der visuellen inertialen Odometrie (VIO) erzielt. Zusätzlich erreicht *x-RIO* sogar auf einem Embedded-Computer Laufzeiten, die um ein vielfaches schneller als Echtzeit sind. Autonome Flüge in Innenräumen und in dichtem Nebel demonstrieren den Praxiseinsatz.

Die Erweiterung Kamera-basierter Ansätze mit Radar-Eigengeschwindigkeitsmessungen nutzt die Vorteile beider Verfahren. So liefert visuelle-inertiale Odometrie sehr gute Ergebnisse, benötigt dafür jedoch gute Sichtbedingungen. Thermale-inertiale Odometrie funktioniert zwar auch bei schlechten Sichtbedingungen, ist jedoch auf Szenen mit Temperaturgradienten angewiesen. Die Bewegungsschätzung mit Radar ist unabhängig beider Einschränkungen einsetzbar, benötigt jedoch ausreichend Objekte innerhalb des Radarscans. Die Kombination dieser Verfahren resultiert in den neuartigen Ansätzen Radar-visuelle-inertiale Odometrie (RVIO) sowie Radar-thermale-inertiale Odometrie (RTIO). Die entwickelte Pipeline *robuste Radar-visuelle/thermale-inertiale Odometrie* (RRxIO) fasst beide Ansätze zusammen. Umfangreiche Evaluation und der Vergleich zu verschiedenen Kombinationen von Ansätzen auf den Radar-thermal-visuell-inertialen Datensätzen zeigt die Vorteile von *RRxIO* deutlich. *RRxIO* erzielt bessere Ergebnisse als *x-RIO* und stellt damit eine weitere Möglichkeit der Navigation in schlechten Sichtbedingungen dar.

Um das Ziel eines globalen Navigationssystems zu erreichen, wurde die Radar-inertiale Navigation um ein Verfahren zur Integration von GNSS-Messungen erweitert. Dies ermöglicht es, die Pose des Fluggeräts auch in urbanen Gebieten absolut in der Welt zu beschreiben. Somit können global geplante Flugpfade auch in schwierigen GNSS-Bedingungen abgeflogen werden. Mit dem vorgestellten Verfahren werden fehlerhafte GNSS-Messungen effektiv von der Fusion ausgeschlossen. Durch die hohe Genauigkeit der Radar-inertialen Navigation werden nur wenige GNSS-Messungen benötigt. Umfangreiche Tests mit getragenen und geflogenen Datensätzen in urbanen Umgebungen einschließlich des Übergangs in Innenräume demonstrieren die Leistungsfähigkeit des Systems. Sogar mit Standard-GPS wurde ein durchschnittlicher globaler Positionsfehler von unter einem Meter in den betrachteten urbanen Datensätzen erzielt.

Zur Kartierung der Umgebung wurde ein Radar-basiertes SLAM-Verfahren entwickelt. Dieses baut auf der Radar-inertialen Navigation auf und nutzt mehrere Radarsensoren gleichzeitig. Ein neuartiges Verfahren ermöglicht die Vorverarbeitung der Radarscans, womit das Ergebnis sowohl der Positionsschätzung als auch der Karte selbst verbessert wird. Schleifenschlüsse werden effektiv erkannt und damit wird ein Positionsdrift unterdrückt. Dies führt auch ohne GNSS-Stützung zu einem langzeitgenauen Navigationssystem. Zusätzlich werden Personen mit einem neuartigen Ansatz im Thermalbild mittels moderner neuronaler Netze detektiert und die Körperhaltung klassifiziert. Eine Echtzeitverarbeitung wird durch den Einsatz von Spezialhardware sichergestellt. Mit Hilfe der Radarscans erfolgt eine präzise Verortung der Personen innerhalb der SLAM-Karte. Die Demonstration eines autonomen Fluges in dichtem Nebel mit online Kartierung unterstreicht die Anwendbarkeit und Leistungsfähigkeit der entwickelten Kartierungspipeline. Hiermit wird das Ziel, die Kartierung auch in schlechten Sichtbedingungen sowie in Echtzeit durchzuführen, erreicht. Im Kontext von Rettungseinsätzen lässt sich dadurch ein umfängliches Lagebild erstellen.

Durch die Entwicklung und Kombination neuartiger Verfahren wurde ein Gesamtsystem entwickelt, das eine zuverlässige Navigation und Kartierung auch unter schwierigen Bedingungen ermöglicht. So wird dies sowohl unter schlechten Sicht- als auch bei schwierigen GNSS-Bedingungen erreicht. Die umfangreiche Evaluation und der Vergleich zu etablierten Verfahren in guten Sichtbedingungen beweist die Leistungsfähigkeit der entwickelten Ansätze. So erzielen diese nicht nur bessere Ergebnisse, sondern können die Navigation auch in schlechten Sichtbedingungen gewährleisten. Das Ziel, ein Navigationssystem aber auch die Kartierung in Echtzeit auf dem Embedded-Computer des Fluggerätes umzusetzen, wurde erreicht. Die Demonstration der online Navigation zeigt die konkrete Anwendbarkeit mit dem Ziel autonome Flüge zu ermöglichen. Somit kann das Szenario in Abbildung 1.1 der Einleitung mit dem vorgeschlagenen System umgesetzt werden. Damit werden neue Möglichkeiten und vielfältige Anwendungsfelder für autonome Fluggeräte eröffnet.

A Anhang

A.1 Mahalanobisdistanz-Test

In Realanwendungen treten neben gültigen Messungen auch Ausreißer, also stark fehlerhaft Messungen auf. Die Fusion dieser kann zu einer enormen Einbuße der Performance oder sogar zum Divergieren des Navigationsfilters führen. Ein häufig genutzte Möglichkeit Ausreißer herauszufiltern besteht in der Anwendung eines Mahalanobis-Distanz-Tests [60]. Die Mahalanobis-Distanz kann mittels [60]

$$d_{\text{Mahalanobis}} = \boldsymbol{r}^T \left(\mathbf{HPH}^T + \mathbf{R}\right)^{-1} \boldsymbol{r} \tag{A.1}$$

berechnet werden. Dabei beschreibt $\boldsymbol{r}$ das Residuum des Filterupdates, $\mathbf{H}$ die Jacobi-Matrix und $\mathbf{R}$ die Messrauschmatrix. Ein Mahalanobis-Distanz-Test vergleicht nun $d_{\text{Mahalanobis}}$ mit einem definierten Perzentil der χ^2-Verteilung mit dem Freiheitsgrad n, was die Schranke d_{thresh} ergibt. Der Freiheitsgrad ergibt sich aus der Länge von r. Die Messung wird als Inlier klassifiziert, falls $d_{\text{Mahalanobis}}$ kleiner als d_{thresh}.

A.2 Mathematik

A.2.1 Schiefsymmetrische Matrix

Die kreuzproduktbildende oder auch schiefsymmetrische Matrix eines dreidimensionalen Vektors $\boldsymbol{a}$ ist durch

$$[\boldsymbol{a}]\times = \begin{bmatrix} 0 & -a_z & a_y \\ a_z & 0 & -a_x \\ -a_y & a_x & 0 \end{bmatrix}. \tag{A.2}$$

definiert. Im Folgenden wird auf wichtige Eigenschaften der schiefsymmetrische Matrix zurückgegriffen. Diese sind die Schiefsymmetrie

$$[\boldsymbol{a}]\times = -\left[\boldsymbol{a}^T\right]\times, \tag{A.3}$$

die Antikommutativität

$$[\boldsymbol{a}] \times \boldsymbol{b} = -[\boldsymbol{b}] \times \boldsymbol{a} \tag{A.4}$$

$$\tag{A.5}$$

und die Definition der Ableitung:

$$\frac{\partial}{\partial \boldsymbol{b}} (\boldsymbol{a} \times \boldsymbol{b}) = [\boldsymbol{a}] \times . \tag{A.6}$$

A.2.2 Fehlerdefinition der Rotationsmatrix

Basierend auf den Fehlerdefinitionen in den Gleichungen (2.1) und (2.11) ergibt sich die Fehlerdefinition der Richtungskosinusmatrix $\mathbf{C}_b^a$ zu:

$$\mathbf{C}_b^a = \left(\mathbf{I}_3 - [\tilde{\boldsymbol{\Psi}}_{ab}]\times\right) \hat{\mathbf{C}}_b^a, \tag{A.7}$$

$$\mathbf{C}_b^{a,T} = \hat{\mathbf{C}}_b^{a,T} \left(\mathbf{I}_3 + [\tilde{\boldsymbol{\Psi}}_{ab}]\times\right). \tag{A.8}$$

A.3 Radar-Geschwindigkeitsupdate

A.3.1 Herleitung der Messmatrizen

Die Herleitung der einzelnen Messmatrizen des Radar-Geschwindigkeitsupdates (5.35) - (5.41) wird im Folgenden für den Fall eines vollständigen Klons für den l-ten Radarzustand durchgeführt. Dabei beziehen sich zur vereinfachten Darstellung alle geschätzten Zustände $\hat{\square}$ auf den geklonten Zustand nach Gleichung (5.24).

Für die Herleitung von $\mathbf{H}_{v,l}$ wird zunächst die Messgleichung mit Hilfe der Fehlerdefinition (2.1) für $\boldsymbol{v}_{ob}^{o}$ zu

$$\tilde{\boldsymbol{v}}_l^r = \hat{\boldsymbol{v}}_l^r - \mathbf{C}_{r,l}^{b,T}\left(\mathbf{C}_{b,l}^{o,T}\boldsymbol{v}_{ob}^o + \left[\boldsymbol{\omega}_{m,l}^b - \boldsymbol{b}_{g,l}^b\right]\times \boldsymbol{l}_{br,l}^b\right) \tag{A.9}$$

$$= \hat{\boldsymbol{v}}_l^r - \mathbf{C}_{r,l}^{b,T}\left(\mathbf{C}_{b,l}^{o,T}\left(\hat{\boldsymbol{v}}_{ob}^o - \tilde{\boldsymbol{v}}_{ob}^o\right) + \left[\boldsymbol{\omega}_{m,l}^b - \boldsymbol{b}_{g,l}^b\right]\times \boldsymbol{l}_{br,l}^b\right) \tag{A.10}$$

umgeformt. Basierend darauf kann die Messmatrix $\mathbf{H}_{v,l}$ durch Linearisierung um den aktuellen Systemzustand $\hat{\boldsymbol{x}}$

$$\mathbf{H}_{v,l} = \left.\frac{\partial \tilde{\boldsymbol{v}}_{ob}^r}{\partial \tilde{\boldsymbol{v}}_{ob}^o}\right|_{\boldsymbol{x}=\hat{\boldsymbol{x}}} = \hat{\mathbf{C}}_{r,l}^{b,T}\hat{\mathbf{C}}_{b,l}^{o,T} \tag{A.11}$$

berechnet werden.

Für $\mathbf{H}_{\boldsymbol{\Psi}_{ob,l}}$ erfolgt analog zunächst die Umformung der Messgleichung mittels der Fehlerdefinition für die Rotationsmatrix (A.8) zu

$$\tilde{\boldsymbol{v}}_l^r = \hat{\boldsymbol{v}}_l^r - \mathbf{C}_{r,l}^{b,T}\left(\mathbf{C}_{b,l}^{o,T}\boldsymbol{v}_{ob}^o + \left[\boldsymbol{\omega}_{m,l}^b - \boldsymbol{b}_{g,l}^b\right]\times \boldsymbol{l}_{br,l}^b\right) \tag{A.12}$$

$$= \hat{\boldsymbol{v}}_l^r - \mathbf{C}_{r,l}^{b,T}\left(\hat{\mathbf{C}}_{b,l}^{o,T}\left(\mathbf{I}_3 + \left[\tilde{\boldsymbol{\Psi}}_{ob}\right]\times\right)\boldsymbol{v}_{ob}^o + \left[\boldsymbol{\omega}_{m,l}^b - \boldsymbol{b}_{g,l}^b\right]\times \boldsymbol{l}_{br,l}^b\right) \tag{A.13}$$

und mit der Vereinfachung für $\boldsymbol{v}_w$ ergibt sich:

$$\tilde{\boldsymbol{v}}_l^r = \hat{\boldsymbol{v}}_l^r - \mathbf{C}_{r,l}^{b,T}\left(\hat{\mathbf{C}}_{b,l}^{o,T}\left(\mathbf{I}_3\boldsymbol{v}_{ob}^o + \left[\tilde{\boldsymbol{\Psi}}_{ob}\right]\times \boldsymbol{v}_{ob}^o\right) + \boldsymbol{v}_{\omega}^b\right), \tag{A.14}$$

was auf Grund der Antikommutativität (A.5) zu

$$\tilde{\boldsymbol{v}}_l^r = \hat{\boldsymbol{v}}_l^r - \mathbf{C}_{r,l}^{b,T} \left(\hat{\mathbf{C}}_{b,l}^{o,T} \left(\mathbf{I}_3 \boldsymbol{v}_{ob}^o - \left[\boldsymbol{v}_{ob}^o \right] \times \tilde{\boldsymbol{\Psi}}_{ob} \right) + \boldsymbol{v}_\omega^b \right) \tag{A.15}$$

$$\tag{A.16}$$

umgeformt werden kann. Für $\mathbf{H}_{\boldsymbol{\Psi}_{ob,l}}$ ergibt sich mit (A.6) durch Linearisierung von (A.15) um den aktuellen Systemzustand $\hat{\boldsymbol{x}}$ zu:

$$\mathbf{H}_{\boldsymbol{\Psi}_{ob,l}} = \left. \frac{\partial \tilde{\boldsymbol{v}}_{ob}^r}{\partial \tilde{\boldsymbol{\Psi}}_{ob}} \right|_{\boldsymbol{x}=\hat{\boldsymbol{x}}} = \hat{\mathbf{C}}_{r,l}^{b,T} \hat{\mathbf{C}}_{b,l}^{o,T} \left[\hat{\boldsymbol{v}}_{ob}^o \right] \times . \tag{A.17}$$

Zur Herleitung von $\mathbf{H}_{bg,l}$ erfolgt wiederum die Umformung der Messgleichung mit der Fehlerdefinition (2.1) für $\boldsymbol{b}_{g,l}^b$:

$$\begin{aligned} \tilde{\boldsymbol{v}}_l^r &= \hat{\boldsymbol{v}}_l^r - \mathbf{C}_{r,l}^{b,T} \left(\mathbf{C}_{b,l}^{o,T} \boldsymbol{v}_{ob}^o + \left[\boldsymbol{\omega}_{m,l}^b - \boldsymbol{b}_{g,l}^b \right] \times \boldsymbol{l}_{br,l}^b \right) && \text{(A.18)} \\ &= \hat{\boldsymbol{v}}_l^r - \mathbf{C}_{r,l}^{b,T} \left(\mathbf{C}_{b,l}^{o,T} \boldsymbol{v}_{ob}^o + \left(\boldsymbol{\omega}_{m,l}^b - \left(\hat{\boldsymbol{b}}_{g,l}^b - \tilde{\boldsymbol{b}}_{g,l}^b \right) \right) \times \boldsymbol{l}_{br,l}^b \right), && \text{(A.19)} \end{aligned}$$

was auf Grund der Antikommutativität (A.5) zu

$$\tilde{\boldsymbol{v}}_l^r = \hat{\boldsymbol{v}}_l^r - \mathbf{C}_{r,l}^{b,T} \left(\mathbf{C}_{b,l}^{o,T} \boldsymbol{v}_{ob}^o - \boldsymbol{l}_{br,l}^b \times \left(\boldsymbol{\omega}_{m,l}^b - \left(\hat{\boldsymbol{b}}_{g,l}^b - \tilde{\boldsymbol{b}}_{g,l}^b \right) \right) \right) \tag{A.20}$$

führt. Die Messmatrix $\mathbf{H}_{bg,l}$ ergibt sich mit (A.6) durch Linearisierung von (A.20) um den aktuellen Systemzustand $\hat{\boldsymbol{x}}$ zu:

$$\mathbf{H}_{bg,l,} = \left. \frac{\partial \tilde{\boldsymbol{v}}_{ob}^r}{\partial \tilde{\boldsymbol{b}}_{g,l}^b} \right|_{\boldsymbol{x}=\hat{\boldsymbol{x}}} = \hat{\mathbf{C}}_{r,l}^{b,T} \left[\hat{\boldsymbol{l}}_{br,l}^b \right] \times \tag{A.21}$$

Für die Herleitung von $\mathbf{H}_{l^b_{br,l}}$ wird wiederum die Messgleichung mit der Fehlerdefinition (2.1), angewendet auf $\boldsymbol{l}^b_{br,l}$, zu

$$\tilde{\boldsymbol{v}}^r_l = \hat{\boldsymbol{v}}^r_l - \mathbf{C}^{b,T}_{r,l}\left(\mathbf{C}^{o,T}_{b,l}\boldsymbol{v}^o_{ob} + \left[\boldsymbol{\omega}^b_{m,l} - \boldsymbol{b}^b_{g,l}\right] \times \boldsymbol{l}^b_{br,l}\right) \tag{A.22}$$

$$= \hat{\boldsymbol{v}}^r_l - \mathbf{C}^{b,T}_{r,l}\left(\boldsymbol{v}^b_{ob} + \left[\boldsymbol{\omega}^b_{m,l} - \boldsymbol{b}^b_{g,l}\right] \times \left(\hat{\boldsymbol{l}}^b_{br,l} - \tilde{\boldsymbol{l}}^b_{br,l}\right)\right) \tag{A.23}$$

umgeformt. Die Messmatrix $\mathbf{H}_{l^b_{br,l}}$ kann mittels (A.6) durch Linearisierung von (A.23) um den aktuellen Systemzustand $\hat{\boldsymbol{x}}$ aufgestellt werden:

$$\mathbf{H}_{l^b_{br,l}} = \left.\frac{\partial \tilde{\boldsymbol{v}}^r_{ob}}{\partial \tilde{\boldsymbol{l}}^b_{br}}\right|_{\boldsymbol{x}=\hat{\boldsymbol{x}}} = \hat{\mathbf{C}}^{b,T}_{r,l}\left[\boldsymbol{\omega}^b_{m,l} - \hat{\boldsymbol{b}}^b_{g,l}\right] \times . \tag{A.24}$$

Abschließend erfolgt die Herleitung von $\mathbf{H}_{\boldsymbol{\Psi}_{br,l}}$ durch Umformung der Messgleichung mittels der Fehlerdefinition für die Rotationsmatrix (A.8), was zu:

$$\tilde{\boldsymbol{v}}^r_l = \hat{\boldsymbol{v}}^r_l - \mathbf{C}^{b,T}_{r,l}\left(\mathbf{C}^{o,T}_{b,l}\boldsymbol{v}^o_{ob} + \left[\boldsymbol{\omega}^b_{m,l} - \boldsymbol{b}^b_{g,l}\right] \times \boldsymbol{l}^b_{br,l}\right) \tag{A.25}$$

$$= \hat{\boldsymbol{v}}^r_l - \left(\hat{\mathbf{C}}^{b,T}_{r,l}\left(\mathbf{I}_3 + \left[\tilde{\boldsymbol{\Psi}}_{br}\right]\times\right)\right)\left(\mathbf{C}^{o,T}_{b,l}\boldsymbol{v}^o_{ob} + \left[\boldsymbol{\omega}^b_{m,l} - \boldsymbol{b}^b_{g,l}\right] \times \boldsymbol{l}^b_{br,l}\right)$$

führt und zu

$$\tilde{\boldsymbol{v}}^r_l = \hat{\boldsymbol{v}}^r_l - \left(\hat{\mathbf{C}}^{b,T}_{r,l}\left(\mathbf{I}_3 + \left[\tilde{\boldsymbol{\Psi}}_{br}\right]\times\right)\right)\left(\boldsymbol{v}^b_{ob} + \boldsymbol{v}^b_{\omega}\right) \tag{A.26}$$

mit $\boldsymbol{v}^b_{ob}$ und $\boldsymbol{v}^b_{\omega}$ vereinfacht werden kann. Ausmultiplizieren ergibt

$$\tilde{\boldsymbol{v}}^r_l = \hat{\boldsymbol{v}}^r_l - \left(\hat{\mathbf{C}}^{b,T}_{r,l}\mathbf{I}_3\left(\boldsymbol{v}^b_{ob} + \boldsymbol{v}^b_{\omega}\right) + \hat{\mathbf{C}}^{b,T}_{r,l}\left[\tilde{\boldsymbol{\Psi}}_{br}\right] \times \left(\boldsymbol{v}^b_{ob} + \boldsymbol{v}^b_{\omega}\right)\right) \tag{A.27}$$

und auf Grund der Antikommutativität (A.5) kann dies zu

$$\tilde{\boldsymbol{v}}^r_l = \hat{\boldsymbol{v}}^r_l - \left(\hat{\mathbf{C}}^{b,T}_{r,l}\mathbf{I}_3\left(\boldsymbol{v}^b_{ob} + \boldsymbol{v}^b_{\omega}\right) - \hat{\mathbf{C}}^{b,T}_{r,l}\left[\boldsymbol{v}^b_{ob} + \boldsymbol{v}^b_{\omega}\right] \times \tilde{\boldsymbol{\Psi}}_{br}\right) \tag{A.28}$$

umgeformt werden Für $\mathbf{H}_{\boldsymbol{\Psi}_{br,l}}$ ergibt sich mit (A.6) durch Linearisierung von (A.28) um den aktuellen Systemzustand $\hat{\boldsymbol{x}}$ zu:

$$\mathbf{H}_{\boldsymbol{\Psi}_{br,l}} = \left.\frac{\partial \tilde{\boldsymbol{v}}^{r}_{ob}}{\partial \tilde{\boldsymbol{\Psi}}^{b}_{r}}\right|_{\boldsymbol{x}=\hat{\boldsymbol{x}}} = \hat{\mathbf{C}}^{b,T}_{r,l} \left[\hat{\boldsymbol{v}}^{b}_{ob} + \hat{\boldsymbol{v}}^{b}_{\omega}\right] \times \tag{A.29}$$

$$= \hat{\mathbf{C}}^{b,T}_{r,l} \left[\hat{\mathbf{C}}^{o,T}_{b,l} \hat{\boldsymbol{v}}^{o}_{ob} + \left[\boldsymbol{\omega}^{b}_{m,l} - \hat{\boldsymbol{b}}^{b}_{g,l}\right] \times \hat{\boldsymbol{l}}^{b}_{br,l}\right] \times . \tag{A.30}$$

A.3.2 Messmatrizen des approximierten Radar-Klons

Analog zu A.3.1 können die Messmatrizen für den approxmierten Klon aufgestellt werden, um das Radar-Geschwindigkeitsupdate aus Abschnitt 5.4.2 auszuführen. Es gilt für die Messmatrix des approximierten Klon des l-ten von L-Radarsensoren $\mathbf{H}_{l,\text{approx}}$:

$$\mathbf{H}_{l,\text{approx}} = \Big[\mathbf{0}_{3\times 12} \quad \mathbf{H}_{l,b_g} \quad \mathbf{0}_{3\times 3} \quad \mathbf{H}_{\text{radar},l} \quad \mathbf{0}_{3\times 9(l-1)} \setminus \\ \mathbf{H}_{l,\text{approx}} \quad \mathbf{0}_{3\times 9(L-l)}\Big] \tag{A.31}$$

$$\mathbf{H}_{l,b_g} = \hat{\mathbf{C}}^{b,T}_{r,l} \left[\hat{\boldsymbol{l}}^{b}_{br,l}\right] \times, \tag{A.32}$$

$$\mathbf{H}_{\text{radar},l} = \left[\mathbf{H}_{l^{b}_{br,l}} \quad \mathbf{H}_{\boldsymbol{\Psi}_{br,l}}\right], \tag{A.33}$$

$$\mathbf{H}_{l^{b}_{r},l} = \hat{\mathbf{C}}^{b,T}_{r,l} \left[\boldsymbol{\omega}^{b}_{m,l}\right] \times, \tag{A.34}$$

$$\mathbf{H}_{\boldsymbol{\Psi}_{br,l}} = \hat{\mathbf{C}}^{b,T}_{r,l} \left[\hat{\mathbf{C}}^{o,T}_{b,l,\text{Klon}} \hat{\boldsymbol{v}}^{n}_{l,\text{Klon}} + [\boldsymbol{\omega}^{b}_{m} - \hat{\boldsymbol{b}}^{b}_{g,l}\right] \times \hat{\boldsymbol{l}}^{b}_{br,l}, \tag{A.35}$$

$$\mathbf{H}_{l,\text{approx}} = \left[\mathbf{0}_{3\times 3} \quad \mathbf{H}_{v,l,\text{Klon}} \quad \mathbf{H}_{\boldsymbol{\Psi}_{ob,l},\text{Klon}}\right], \tag{A.36}$$

$$\mathbf{H}_{v,l\text{Klon}} = \hat{\mathbf{C}}^{b,T}_{r,l} \hat{\mathbf{C}}^{o,T}_{b,l,\text{Klon}}, \tag{A.37}$$

$$\mathbf{H}_{\boldsymbol{\Psi}_{ob,l,\text{Klon}}} = \hat{\mathbf{C}}^{b,T}_{r,l} \left[\hat{\mathbf{C}}^{o,T}_{b,l,\text{Klon}} \hat{\boldsymbol{v}}^{o}_{ob,\text{Klon}} + \left[\boldsymbol{\omega}^{b}_{m,l} - \hat{\boldsymbol{b}}^{b}_{g,l}\right] \times \hat{\boldsymbol{l}}^{b}_{br,l}\right] \times . \tag{A.38}$$

A.4 Simulationsergebnisse der Radar-inertialen Navigation

Die Darstellung aller drei extrinsichen Radar-Zustände der simulationen Evaluation der Radar-inertialen Navigation aus Abschnitt 5.4.5 ist in Abbildung A.1 für das Szenario *niedrige-Dynamik* und in Abbildung A.2 gegeben.

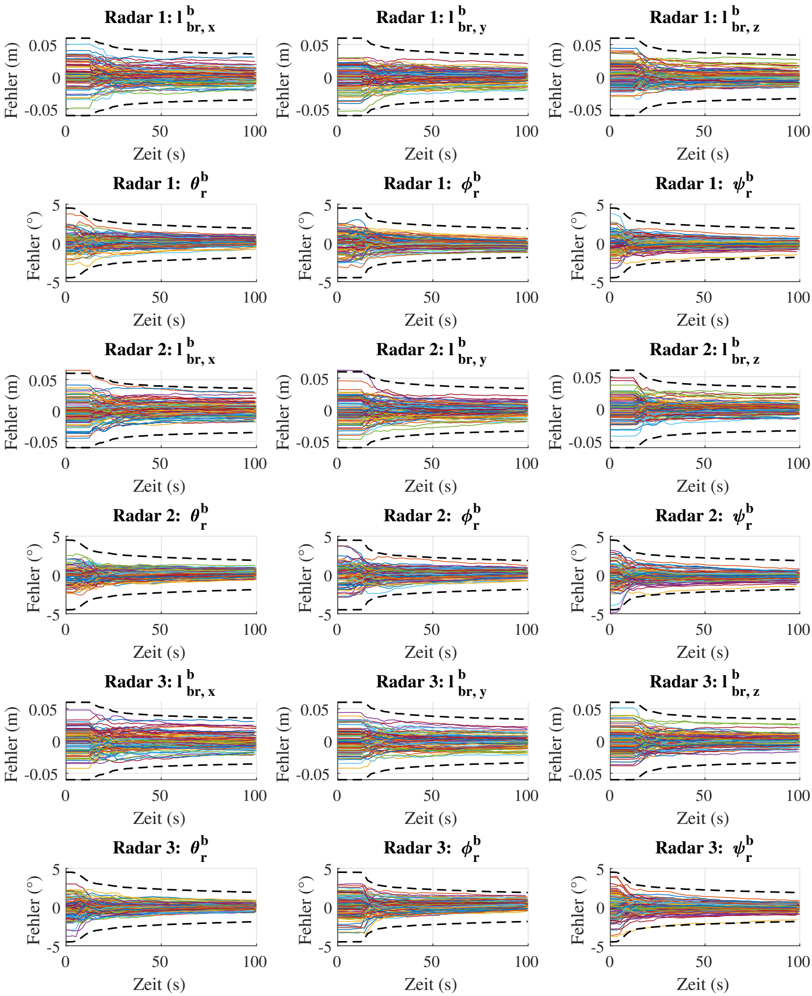


Abbildung A.1: Ergebnis der Monte-Carlo Simulation im Szenario *niedrige-Dynamik* für die Schätzung der extrinsischen Kalibrierung der Translation $\boldsymbol{l}^b_{br}$ und der Rotation $\boldsymbol{q}^b_r$ bei einem Setup von drei Radarsensoren. Der geschätzte $\pm 3\sigma$ Schlauch ist schwarz gestrichelt.

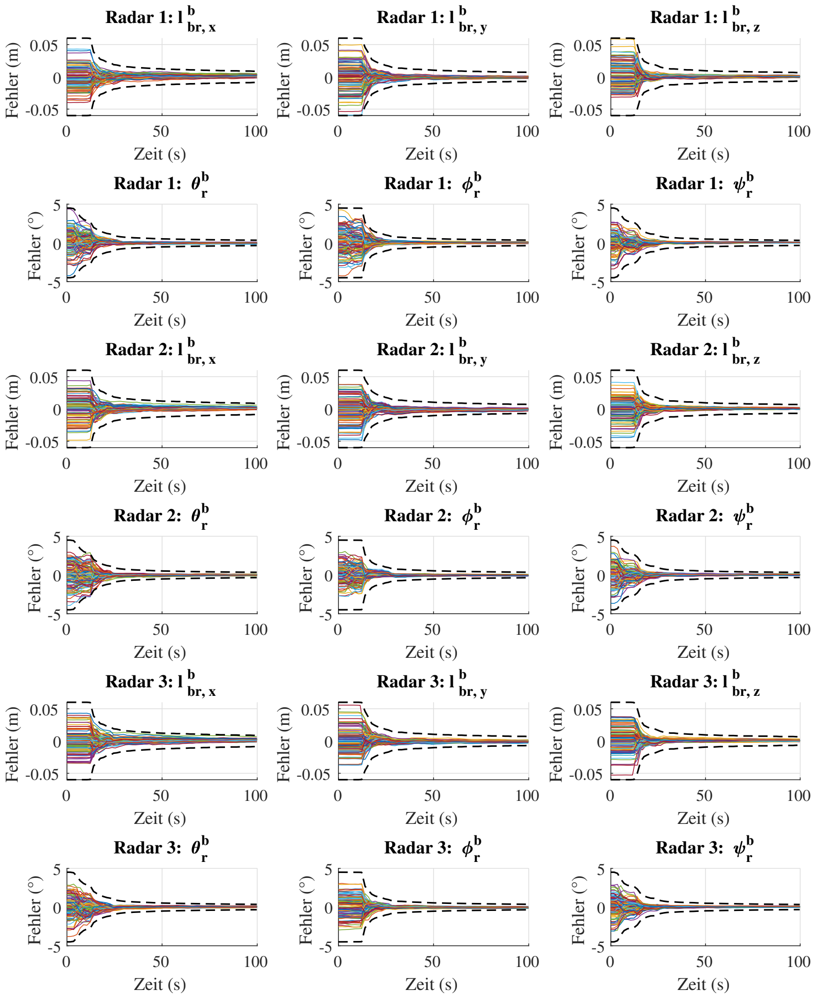


Abbildung A.2: Ergebnis der Monte-Carlo Simulation im Szenario *hohe-Dynamig* für die Schätzung der extrinsischen Kalibrierung der Translation $\boldsymbol{l}_{br}^{b}$ und der Rotation $\boldsymbol{q}_{r}^{b}$ bei einem Setup von drei Radarsensoren. Der geschätzte $\pm 3\sigma$ Schlauch ist schwarz gestrichelt.

Literaturverzeichnis

Quellenangaben

[1] *CPU Benchmark*. `https://www.cpubenchmark.net/`.

[2] *Global Positioning Systems Directorate Systems Engineering and Integration, Ed.* Navstar GPS Space Segment/Navigation User Interfaces, 2019.

[3] M. Abadi, A. Agarwal, P. Barham, E. Brevdo, Z. Chen, C. Citro, G. S. Corrado, A. Davis, J. Dean, M. Devin, S. Ghemawat, I. Goodfellow, A. Harp, G. Irving, M. Isard, Y. Jia, R. Jozefowicz, L. Kaiser, M. Kudlur, J. Levenberg, D. Mané, R. Monga, S. Moore, D. Murray, C. Olah, M. Schuster, J. Shlens, B. Steiner, I. Sutskever, K. Talwar, P. Tucker, V. Vanhoucke, V. Vasudevan, F. Viégas, O. Vinyals, P. Warden, M. Wattenberg, M. Wicke, Y. Yu und X. Zheng: *TensorFlow: Large-Scale Machine Learning on Heterogeneous Systems*, 2015. `https://www.tensorflow.org/`, Software available from tensorflow.org.

[4] R. Aldera, D. De Martini, M. Gadd und P. Newman: *Fast Radar Motion Estimation with a Learnt Focus of Attention using Weak Supervision*. In: *2019 International Conference on Robotics and Automation (ICRA)*, S. 1190–1196. 2019.

[5] R. Aldera, D. De Martini, M. Gadd und P. Newman: *What Could Go Wrong? Introspective Radar Odometry in Challenging Environments*. In: *2019 IEEE Intelligent Transportation Systems (ITSC)*. 2019.

[6] Y. Almalioglu, M. Turan, C. X. Lu, N. Trigoni und A. Markham: *Milli-RIO: Ego-Motion Estimation With Low-Cost Millimetre-Wave Radar*. IEEE Sensors Journal, 21(3):3314–3323, 2021.

[7] Amazon: *Amazon Prime Air*. `https://www.aboutamazon.de/news/innovationen/prime-air`.

[8] J. Atman: *Strategien zur Lokalisierung und semantische Kartierung für mobile multisensorielle Plattformen*. Dissertation, Karlsruher Institut für Technologie (KIT), 2021.

[9] M. Barjenbruch, D. Kellner, J. Klappstein, J. Dickmann und K. Dietmayer: *Joint spatial- and Doppler-based ego-motion estimation for automotive radars*. In: *2015 IEEE Intelligent Vehicles Symposium (IV)*, S. 839–844, 2015.

[10] J. L. Barron, R. E. Mercer, X. Chen und P. Joe: *3d velocity from 3d doppler radial velocity*. International Journal of Imaging Systems and Technology, 15(3):189–198, 2005.

[11] J. Berni, P. Zarco-Tejada, L. Suárez, V. González-Dugo und E. Fereres: *Remote sensing of vegetation from UAV platforms using lightweight multispectral and thermal imaging sensors*. Int. Arch. Photogramm. Remote Sens. Spatial Inform. Sci, 38(6):6, 2009.

[12] M. Bloesch, M. Burri, S. Omari, M. Hutter und R. Siegwart: *Iterated extended Kalman filter based visual-inertial odometry using direct photometric feedback*. The International Journal of Robotics Research, 36, 2017.

[13] P. T. Boggs und J. E. Rogers: *Orthogonal distance regression*. Contemporary Mathematics, 112:183–194, 1990.

[14] P. V. K. Borges und S. Vidas: *Practical Infrared Visual Odometry*. IEEE Transactions on Intelligent Transportation Systems, 17(8):2205–2213, 2016.

[15] G. Bradski: *The OpenCV Library*. Dr. Dobb's Journal of Software Tools, 2000.

[16] L. Breiman: *Random Forests*. Machine Learning, 45(1):5–32, 2001.

[17] M. Burri, J. Nikolic, P. Gohl, T. Schneider, J. Rehder, S. Omari, M. W. Achtelik und R. Siegwart: *The EuRoC micro aerial vehicle datasets*. The International Journal of Robotics Research, 2016.

[18] C. Campos, R. Elvira, J. J. G. Rodríguez, J. M. M. Montiel und J. D. Tardós: *ORB-SLAM3: An Accurate Open-Source Library for Visual, Visual–Inertial, and Multimap SLAM*. IEEE Transactions on Robotics, 37(6):1874–1890, 2021.

[19] Z. Cao, G. Hidalgo, T. Simon, S. E. Wei und Y. Sheikh: *OpenPose: Realtime Multi-Person 2D Pose Estimation using Part Affinity Fields*. Dez. 2018.

[20] S. H. Cen und P. Newman: *Precise Ego-Motion Estimation with Millimeter-Wave Radar Under Diverse and Challenging Conditions*. In: *2018 IEEE International Conference on Robotics and Automation (ICRA)*, S. 1–8. IEEE, 2018.

[21] I. C. Chen, C. J. Wang, C. K. Wen und S. J. Tzou: *Multi-Person Pose Estimation Using Thermal Images*. IEEE Access, 8:174964–174971, 2020.

[22] X. Chen, J. L. Barron, R. E. Mercer und P. Joe: *3D regularized velocity from 3D Doppler radial velocity*. In: *Proceedings 2001 International Conference on Image Processing*, Bd. 3, S. 664–667 vol.3, 2001.

[23] Y. Chen, Y. Tian und M. He: *Monocular human pose estimation: A survey of deep learning-based methods*. Computer Vision and Image Understanding, 192:102897, 2020.

[24] K. Cisek, A. Zolich, K. Klausen und T. A. Johansen: *Ultra-wide band Real time Location Systems: Practical implementation and UAV performance evaluation*. In: *2017 Workshop on Research, Education and Development of Unmanned Aerial Systems (RED-UAS)*, S. 204–209, 2017.

[25] M. Contributors: *OpenMMLab Pose Estimation Toolbox and Benchmark*. `https://github.com/open-mmlab/mmpose`, 2020.

[26] Coral: *Benchmark Edge TPU Modelle*. `https://coral.ai/models/object-detection/`.

[27] Coral: *Benchmark Edge TPU Modelle*. `https://coral.ai/models/pose-estimation/`.

[28] Coral: *Next-Generation Pose Detection with MoveNet and TensorFlow.js*. `https://blog.tensorflow.org/2021/05/next-generation-pose-detection-with-movenet-and-tensorflowjs.html`.

[29] J. M. Coughlan und A. L. Yuille: *Manhattan World: compass direction from a single image by Bayesian inference*. In: *Proceedings of the Seventh IEEE International Conference on Computer Vision*, Bd. 2, S. 941–947 vol.2, 1999.

[30] P. Dabove, V. Di Pietra, M. Piras, A. A. Jabbar und S. A. Kazim: *Indoor positioning using Ultra-wide band (UWB) technologies: Positioning accuracies and sensors' performances*. In: *2018 IEEE/ION Position, Location and Navigation Symposium (PLANS)*, S. 175–184, April 2018.

[31] J. Dai, Y. Li, K. He und J. Sun: *R-FCN: Object Detection via Region-based Fully Convolutional Networks.* CoRR, abs/1605.06409, 2016.

[32] J. Delaune, R. Hewitt, L. Lytle, C. Sorice, R. Thakker und L. Matthies: *Thermal-Inertial Odometry for Autonomous Flight Throughout the Night*. In: *2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, S. 1122–1128.

[33] J. Delmerico und D. Scaramuzza: *A Benchmark Comparison of Monocular Visual-Inertial Odometry Algorithms for Flying Robots*. In: *2018 IEEE International Conference on Robotics and Automation (ICRA)*, S. 2502–2509, 2018.

[34] J. Dickmann, J. Klappstein, M. Hahn, N. Appenrodt, H. Bloecher, K. Werber und A. Sailer: *Äutomotive radar the key technology for autonomous driving: From detection and ranging to environmental understanding"*. In: *2016 IEEE Radar Conference (RadarConf)*, S. 1–6, 2016.

[35] DJI: *Technische Daten der Mavic-3*. `https://www.dji.com/de/mavic-3/specs`.

[36] Y. Dobrev, Y. Dobrev, P. Gulden, M. Lipka, T. Pavlenko, D. Moormann und M. Vossiek: *Radar-Based High-Accuracy 3D Localization of UAVs for Landing in GNSS-Denied Environments*. In: *2018 IEEE MTT-S International Conference on Microwaves for Intelligent Mobility (ICMIM)*, S. 1–4, April 2018.

[37] Y. Dobrev, M. Schuett, P. Hartmann und D. Moormann: *Deutscher Luft- und Raumfahrtkongress 2016, Institut für Flugsystemdynamik, RWTH Aachen University, Germany*. In: *Entwurf und Validierung eines Praezisionslandesystems für unbemannte Tiltwing-Fluggeräte*, 2016.

[38] P. Doherty und P. Rudol: *A UAV Search and Rescue Scenario with Human Body Detection and Geolocalization*. In: M. A. Orgun und J. Thornton (Hrsg.): *AI 2007: Advances in Artificial Intelligence*, S. 1–13, Berlin, Heidelberg, 2007. Springer Berlin Heidelberg.

[39] EASA - European Union Aviation Safety Agency. In: *Certification Specifications and Guidance Material for the design of surface-level VFR heliports located at aerodromes that fall under the scope of Regulation (EU) 2018/1139(CS-HPT-DSN)*, Bd. Issue 1, 2019.

[40] M. A. Fischler und R. C. Bolles: *Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography*. Communications of the ACM, 1981.

[41] A. Flint, D. Murray und I. Reid: *Manhattan scene understanding using monocular, stereo, and 3D features*. In: *2011 International Conference on Computer Vision*, S. 2228–2235, 2011.

[42] B. für Bildung und Forschung: *Flugsystem-Assistierte Leitung komplexer Einsatzlagen - FALKE*. https://www.sifo.de/sifo/de/projekte/querschnittsthemen-und-aktivitaeten/praxistransfer-und-kompetenzaufbau/kmu-innovativ/falke/flugsystem-assistierte-leitung-komplexer-einsatzlagen.html.

[43] C. Forster, L. Carlone, F. Dellaert und D. Scaramuzza: *On-Manifold Preintegration for Real-Time Visual–Inertial Odometry*. IEEE Transactions on Robotics, 33(1):1–21, 2017.

[44] P. Furgale, J. Rehder und R. Siegwart: *Unified temporal and spatial calibration for multi-sensor systems*. In: *2013 IEEE/RSJ International Conference on Intelligent Robots and Systems*, S. 1280–1286, 2013.

[45] A. Geiger, P. Lenz und R. Urtasun: *Are we ready for Autonomous Driving? The KITTI Vision Benchmark Suite*. In: *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2012.

[46] P. Geneva, K. Eckenhoff, W. Lee, Y. Yang und G. Huang: *OpenVINS: A Research Platform for Visual-Inertial Estimation*. In: *2020 IEEE International Conference on Robotics and Automation (ICRA)*, S. 4666–4672, 2020.

[47] P. Geneva und G. Huang: *vicon2gt: Derivations and Analysis*. Techn. Ber. RPNG-2020-VICON2GT, University of Delaware, 2020. Available: http://udel.edu/~ghuang/papers/tr_vicon2gt.pdf.

[48] A. Geron: *Hands-on machine learning with Scikit-Learn and TensorFlow : concepts, tools, and techniques to build intelligent systems*. O'Reilly Media, Sebastopol, CA, 2017.

[49] R. Ghabcheloo und S. Siddiqui: *Complete Odometry Estimation of a Vehicle Using Single Automotive Radar and a Gyroscope*. In: *2018 26th Mediterranean Conference on Control and Automation (MED)*, S. 855–860. IEEE, 19.06.2018 - 22.06.2018.

[50] S. Ghazal und U. S. Khan: *Human posture classification using skeleton information*. In: *2018 International Conference on Computing, Mathematics and Engineering Technologies (iCoMET)*, S. 1–4, 2018.

[51] C. D. Ghilani: *Adjustment computations: Spatial data analysis / Charles D. Ghilani*. John Wiley & Sons, Hoboken, New Jersey, 2017.

[52] A. Goodchild und J. Toy: *Delivery by drone: An evaluation of unmanned aerial vehicle technology in reducing CO2 emissions in the delivery service industry*. Transportation Research Part D: Transport and Environment, 61:58 – 67, 2018.

[53] P. Groves: *Principles of GNSS, Inertial, and Multisensor Integrated Navigation Systems, Second Edition*. 2013.

[54] A. Gupta, K. Gupta, K. N. M. Gupta und K. O. Gupta: *Human Activity Recognition Using Pose Estimation and Machine Learning Algorithm*. In: *ISIC*, 2021.

[55] W. Hess, D. Kohler, H. Rapp und D. Andor: *Real-time loop closure in 2D LIDAR SLAM*. In: *2016 IEEE International Conference on Robotics and Automation (ICRA)*, S. 1271–1278, 2016.

[56] M. Holder, S. Hellwig und H. Winner: *Real-Time Pose Graph SLAM based on Radar*. In: *2019 IEEE Intelligent Vehicles Symposium (IV)*, S. 1145–1151, 2019.

[57] P. Huegler, B. Driemeyer, T. Chaloun und C. Waldschmidt: *122 GHz Monostatic Radar Altimeter for Automated UAV Landing*. In: *2018 International Conference on Electromagnetics in Advanced Applications (ICEAA)*, S. 161–164, Sep. 2018.

[58] P. Huegler, M. Geiger und C. Waldschmidt: *77 GHz radar-based altimeter for unmanned aerial vehicles*. In: *2018 IEEE Radio and Wireless Symposium (RWS)*, S. 129–132. IEEE, 2018.

[59] C. Iovescu und S. Rao: *The fundamentals of millimeter wave sensors*. Texas Instruments, 2017.

[60] A. J. Izenman: *Modern Multivariate Statistical Techniques: Regression, Classification, and Manifold Learning*. Springer Publishing Company, Incorporated, 1. Aufl., 2008.

[61] M. Jankiraman: *FMCW radar design*. Artech House radar series. Artech House, Norwood, MA, 2018.

[62] Y. Jung, H. Bang und D. Lee: *Robust marker tracking algorithm for precise UAV vision-based autonomous landing*. In: *2015 15th International Conference on Control, Automation and Systems (ICCAS)*, S. 443–446, Oct 2015.

[63] R. E. Kalman und Others: *A new approach to linear filtering and prediction problems*. Journal of basic Engineering, 82(1):35–45, 1960.

[64] E. Kaplan: *Understanding GPS - Principles and applications*. Artech House, 2005.

[65] Y. Karaca, M. Cicek, O. Tatli, A. Sahin, S. Pasli, M. F. Beser und S. Turedi: *The potential use of unmanned aircraft systems (drones) in mountain search and rescue operations*. The American Journal of Emergency Medicine, 36(4):583 – 588, 2018.

[66] D. Kellner, M. Barjenbruch, K. Dietmayer, J. Klappstein und J. Dickmann: *Joint radar alignment and odometry calibration*. In: *2015 18th International Conference on Information Fusion (Fusion)*, S. 366–374, 2015.

[67] D. Kellner, M. Barjenbruch, J. Klappstein, J. Dickmann und K. Dietmayer: *Instantaneous ego-motion estimation using Doppler radar*. In: *Conference on Intelligent Transportation Systems (ITSC 2013)*, S. 869–874. IEEE, 2013.

[68] D. Kellner, M. Barjenbruch, J. Klappstein, J. Dickmann und K. Dietmayer: *Instantaneous ego-motion estimation using multiple Doppler radars*. In: *2014 International Conference on Robotics and Automation (ICRA)*, S. 1592–1597, 2014.

[69] S. Khattak, F. Mascarich, T. Dang, C. Papachristos und K. Alexis: *Robust Thermal-Inertial Localization for Aerial Robots: A Case for Direct Methods*. In: *2019 International Conference on Unmanned Aircraft Systems (ICUAS)*.

[70] S. Khattak, C. Papachristos und K. Alexis: *Marker Based Thermal-Inertial Localization for Aerial Robots in Obscurant Filled Environments*. In: *Advances in Visual Computing*, S. 565–575, Cham, 2018. Springer International Publishing.

[71] S. Khattak, C. Papachristos und K. Alexis: *Visual-Thermal Landmarks and Inertial Fusion for Navigation in Degraded Visual Environments*. In: *2019 IEEE Aerospace Conference*, S. 1–9, 2019.

[72] S. Khattak, C. Papachristos und K. Alexis: *Keyframe-based thermal–inertial odometry*, 2020.

[73] N. Koenig und A. Howard: *Design and use paradigms for Gazebo, an open-source multi-robot simulator*. In: *2004 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS) (IEEE Cat. No.04CH37566)*, Bd. 3, S. 2149–2154 vol.3, 2004.

[74] S. Kohlbrecher, J. Meyer, O. von Stryk und U. Klingauf: *A Flexible and Scalable SLAM System with Full 3D Motion Estimation*. In: *Proc. IEEE International Symposium on Safety, Security and Rescue Robotics (SSRR)*. IEEE, November 2011.

[75] T. Krajník, M. Nitsche, J. Faigl, T. Duckett, M. Mejail und L. Přeučil: *External localization system for mobile robotics*. In: *16th International Conference on Advanced Robotics (ICAR)*, Nov 2013.

[76] A. Kramer, C. Stahoviak, A. Santamaria-Navarro, A. a. Aghamohammadi und C. Heckman: *Radar-Inertial Ego-Velocity Estimation for Visually Degraded Environments*. In: *To be published at 2020 IEEE International Conference on Robotics and Automation*. IEEE, 2020.

[77] S. Kreiss, L. Bertoni und A. Alahi: *PifPaf: Composite Fields for Human Pose Estimation*. März 2019.

[78] H. Kuusniemi: *User level reliability and quality monitoring in satellite based personal navigation*, 2005.

[79] Laenderdaten.info: *Durchschnittsgröße von Mann und Frau*. `https://www.laenderdaten.info/durchschnittliche-koerpergroessen.php`.

[80] S. Leutenegger, S. Lynen, M. Bosse, R. Siegwart und P. Furgale: *Keyframe-based visual–inertial odometry using nonlinear optimization*. The International Journal of Robotics Research, 34(3):314–334, 2015.

[81] M. Li, H. Yu, X. Zheng und A. I. Mourikis: *High-fidelity sensor modeling and self-calibration in vision-aided inertial navigation*. In: *2014 IEEE International Conference on Robotics and Automation (ICRA)*, S. 409–416. IEEE, 2014.

[82] Y. Liang, S. Mueller, D. Schwendner, D. Rolle, D. Ganesch und I. Schaffer: *A Scalable Framework for Robust Vehicle State Estimation with a Fusion of a Low-Cost IMU, the GNSS, Radar, a Camera and Lidar*.

In: *2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, S. 1661–1668, 2020.

[83] T. Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollár und C. L. Zitnick: *Microsoft COCO: Common Objects in Context*. In: D. Fleet, T. Pajdla, B. Schiele und T. Tuytelaars (Hrsg.): *Computer Vision – ECCV 2014*, S. 740–755, Cham, 2014. Springer International Publishing.

[84] S. S. Liu, M. Atia, T. T. B. Karamat, A. A. Noureldin und S. S. Givigi: *LiDAR-aided integrated INS/GPS navigation system for Unmanned Ground vehicles in urban and indoor environments using hybrid adaptive scan matching algorithm*. In: *28th International Technical Meeting of the Satellite Division of the Institute of Navigation, ION GNSS 2015*, S. 2311–2318, Jan. 2015.

[85] C. X. Lu, S. Rosa, P. Zhao, B. Wang, C. Chen, J. A. Stankovic, N. Trigoni und A. Markham: *See through Smoke: Robust Indoor Mapping with Low-Cost MmWave Radar*. In: *Proceedings of the 18th International Conference on Mobile Systems, Applications, and Services*, MobiSys '20, S. 14–27. Association for Computing Machinery, 2020.

[86] C. X. Lu, M. R. U. Saputra, P. Zhao, Y. Almalioglu, P. P. B. de Gusmao, C. Chen, K. Sun, N. Trigoni und A. Markham: *MilliEgo: Single-Chip MmWave Radar Aided Egomotion Estimation via Deep Sensor Fusion*. In: *Proceedings of the 18th Conference on Embedded Networked Sensor Systems*, SenSys '20, 2020.

[87] V. U. Luftfahren: *Analyse des deutschen Drohnenmarktes 2021*. `https://www.verband-unbemannte-luftfahrt.de/wp-content/uploads/2021/04/VUL-Markststudie_2021_DE_f_April-2021.pdf`.

[88] S. Lupfer, M. Rapp, K. Dietmayer, P. Brosseit, J. Lombacher, M. Hahn und J. Dickmann: *Increasing FastSLAM accuracy for radar data by integrating the Doppler information*. In: *2017 IEEE MTT-S International Conference on Microwaves for Intelligent Mobility (ICMIM)*, S. 103–106, 2017.

[89] S. Lynen, M. W. Achtelik, S. Weiss, M. Chli und R. Siegwart: *A robust and modular multi-sensor fusion approach applied to MAV navigation*. In: *2013 IEEE/RSJ International Conference on Intelligent Robots and Systems*, S. 3923–3929, 2013.

[90] J. Mallon und P. F. Whelan: *Which pattern? Biasing aspects of planar calibration patterns and detection methods*. Pattern Recognition Letters, 28(8):921–930, 2007.

[91] R. Mark A., S. James A. und H. William A.: *Principles of Modern Radar : Basic Principles, Volume 1*. Nr. Volume I, Basic principles. SciTech Publishing, 2010.

[92] A. Marut, K. Wojtowicz und K. Falkowski: *ArUco markers pose estimation in UAV landing aid system*. In: *2019 IEEE 5th International Workshop on Metrology for AeroSpace (MetroAeroSpace)*, S. 261–266, June 2019.

[93] R. Mascaro, L. Teixeira, T. Hinzmann, R. Siegwart und M. Chli: *GOMSF: Graph-Optimization Based Multi-Sensor Fusion for robust UAV Pose estimation*. In: *2018 IEEE International Conference on Robotics and Automation (ICRA)*, S. 1421–1428, 2018.

[94] X. Meng, H. Wang und B. Liu: *A Robust Vehicle Localization Approach Based on GNSS/IMU/DMI/LiDAR Sensor Fusion for Autonomous Vehicles*. Sensors, 17(9), 2017.

[95] C. D. Monaco und S. Brennan: *RADARODO: Ego-Motion Estimation from Doppler and Spatial Data in RADAR Images*. IEEE Transactions on Intelligent Vehicles, S. 1–1, 2020.

[96] M. Montemerlo, S. Thrun, D. Roller und B. Wegbreit: *FastSLAM 2.0: An Improved Particle Filtering Algorithm for Simultaneous Localization and Mapping That Provably Converges*. In: *Proceedings of the 18th International Joint Conference on Artificial Intelligence*, IJCAI'03, S. 1151–1156, San Francisco, CA, USA, 2003. Morgan Kaufmann Publishers Inc.

[97] M. Mostafa, S. Zahran, A. Moussa, N. El-Sheimy und A. Sesay: *Radar and Visual Odometry Integrated System Aided Navigation for UAVS in GNSS Denied Environment*. Sensors, 18(9), 2018.

[98] A. I. Mourikis und S. I. Roumeliotis: *A Multi-State Constraint Kalman Filter for Vision-aided Inertial Navigation*. IEEE Transactions on Robotics, S. 3565–3572, 2007.

[99] K. Mueller, J. Atman, N. Kronenwett und G. F. Trommer: *A Multi-Sensor Navigation System for Outdoor and Indoor Environments*. In: *Proceedings of the 2020 International Technical Meeting of The Institute of Navigation*, 2020.

[100] M. W. Mueller, M. Hehn und R. D'Andrea: *A Computationally Efficient Motion Primitive for Quadrocopter Trajectory Generation*. IEEE Transactions on Robotics, 31(6):1294–1310, 2015.

[101] Z. Ning und H. Hu: *An Experimental Study on the Aerodynamic and Aeroacoustic Performances of a Bio-Inspired UAV Propeller*. Juni 2017.

[102] J. Nocedal und S. J. Wright: *Numerical optimization*. Springer series in operations research and financial engineering. Springer, New York, 2nd ed. Aufl., 2006.

[103] Novatel. In: *NovAtel CORRECT - Optimize your positioning performance*, Nr. https://www.novatel.com/assets/Documents/Papers/NovAtelCORRECT-Brochure.pdf, December 2016.

[104] A. Otto, N. Agatz, J. Campbell, B. Golden und E. Pesch: *Optimization approaches for civil applications of unmanned aerial vehicles (UAVs) or aerial drones: A survey*. Networks, 72(4):411–458, 2018.

[105] P. E. Pace: *Detecting and classifying low probability of intercept radar*. Artech House, 2009.

[106] G. Papandreou, T. Zhu, L. C. Chen, S. Gidaris, J. Tompson und K. Murphy: *PersonLab: Person pose estimation and instance segmentation with a bottom-up, part-based, geometric embedding model*. In: *Proceedings of the European Conference on Computer Vision (ECCV)*, S. 269–286, 2018.

[107] Y. S. Park, Y. S. Shin und A. Kim: *PhaRaO: Direct Radar Odometry using Phase Correlation*. In: *2020 IEEE International Conference on Robotics and Automation (ICRA)*, S. 2617–2623, 2020.

[108] J. Patterson und A. Gibson: *Deep Learning: A Practitioner's Approach*. O'Reilly, Beijing, 2017.

[109] D. Peisen und B. Sawyer: *Heliport/Vertiport MLS Precision Approaches*. In: *SAIC Systems Control Technology Inc. and U.S. Department of Transportation, Federal Aviation Administration*, 1994.

[110] T. Qin, P. Li und S. Shen: *VINS-Mono: A Robust and Versatile Monocular Visual-Inertial State Estimator*. IEEE Transactions on Robotics, 34(4):1004–1020, 2018.

[111] E. B. Quist und R. W. Beard: *Radar odometry on fixed-wing small unmanned aircraft*. IEEE Transactions on Aerospace and Electronic Systems, 52(1):396–410, 2016.

[112] E. B. Quist, P. C. Niedfeldt und R. W. Beard: *Radar odometry with recursive-RANSAC*. IEEE Transactions on Aerospace and Electronic Systems, 52(4):1618–1630, 2016.

[113] M. Rapp, M. Barjenbruch, K. Dietmayer, M. Hahn und J. Dickmann: *A fast probabilistic ego-motion estimation framework for radar*. In: *2015 European Conference on Mobile Robots (ECMR)*, S. 1–6, 2015.

[114] J. Redmon und A. Farhadi: *YOLOv3: An Incremental Improvement*. Apr. 2018.

[115] S. Ren, K. He, R. B. Girshick und J. Sun: *Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks.* In: C. Cortes, N. D. Lawrence, D. D. Lee, M. Sugiyama und R. Garnett (Hrsg.): *NIPS*, S. 91–99, 2015.

[116] F. Romero-Ramirez, R. Muñoz-Salinas und R. Medina-Carnicer: *Speeded Up Detection of Squared Fiducial Markers*. Image and Vision Computing, 76, 2018.

[117] F. Roos, J. Bechter, C. Knill, B. Schweizer und C. Waldschmidt: *Radar Sensors for Autonomous Driving: Modulation Schemes and Interference Mitigation*. IEEE Microwave Magazine, 20(9):58–72, 2019.

[118] A. Rosebrock: *Intersection over Union (IoU) for object detection*. `https://pyimagesearch.com/2016/11/07/intersection-over-union-iou-for-object-detection/`.

[119] J. C. Rosser, V. Vignesh, B. A. Terwilliger und B. C. Parker: *Surgical and Medical Applications of Drones: A Comprehensive Review*. JSLS : Journal of the Society of Laparoendoscopic Surgeons, 22(3), 2018.

[120] T. Rosten, Edwardand Drummond: *Machine Learning for High-Speed Corner Detection*. In: A. Leonardis, H. Bischof und A. Pinz (Hrsg.): *Computer Vision – ECCV 2006*, 2006.

[121] S. I. Roumeliotis und J. W. Burdick: *Stochastic cloning: A generalized framework for processing relative state measurements*. In: *Proceedings 2002 IEEE International Conference on Robotics and Automation (Cat. No.02CH37292)*, Bd. 2, S. 1788–1795 vol.2, 2002.

[122] G. Ruck: *Radar cross section handbook*. Plenum Press, 1970.

[123] R. S.: *Computer vision: algorithms and applications*. Springer Science & Business Media, 2010.

[124] F. F. Sales, D. Portugal und R. P. Rocha: *Real-time people detection and mapping system for a mobile robot using a RGB-D sensor*. In: *2014 11th International Conference on Informatics in Control, Automation and Robotics (ICINCO)*, Bd. 02, S. 467–474, 2014.

[125] M. Sandler, A. Howard, M. Zhu, A. Zhmoginov und L. C. Chen: *MobileNetV2: Inverted Residuals and Linear Bottlenecks*, 2018. cite arxiv:1801.04381.

[126] M. R. U. Saputra, P. P. B. de Gusmao, C. X. Lu, Y. Almalioglu, S. Rosa, C. Chen, J. Wahlstroem, W. Wang, A. Markham und N. Trigoni: *DeepTIO: A Deep Thermal-Inertial Odometry With Visual Hallucination*. IEEE Robotics and Automation Letters, 5(2):1672–1679, 2020.

[127] A. F. Scannapieco, A. Renga und A. Moccia: *Investigation on radar-based applications for mini-UAS and MAVs*. In: *2016 17th International Radar Symposium (IRS)*, S. 1–6, 2016.

[128] D. Scaramuzza, M. C. Achtelik, L. Doitsidis, F. Friedrich, E. Kosmatopoulos, A. Martinelli, M. W. Achtelik, M. Chli, S. Chatzichristofis, L. Kneip, D. Gurdan, L. Heng, G. H. Lee, S. Lynen, M. Pollefeys, A. Renzaglia, R. Siegwart, J. C. Stumpf, P. Tanskanen, C. Troiani, S. Weiss und L. Meier: *Vision-Controlled Micro Flying Robots: From System Design to Autonomous Navigation and Mapping in GPS-Denied Environments*. IEEE Robotics Automation Magazine, 21(3):26–40, 2014.

[129] M. Schartel, R. Burr, P. Schoeder, G. Rossi, P. Hügler, W. Mayer und C. Waldschmidt: *Radar-based altitude over ground estimation of UAVs*. In: *2018 11th German Microwave Conference (GeMiC)*, S. 103–106, March 2018.

[130] D. Schleicher, L. M. Bergasa, M. Ocana, R. Barea und E. Lopez: *Real-time hierarchical GPS aided visual SLAM on urban environments*. In: *2009 IEEE International Conference on Robotics and Automation*, S. 4381–4386, 2009.

[131] F. Schuster, C. G. Keller, M. Rapp, M. Haueis und C. Curio: *Landmark based radar SLAM using graph optimization*. In: *2016 IEEE International Conference on Intelligent Transportation Systems (ITSC)*, S. 2559–2564. 2016.

[132] J. E. Scott und C. H. Scott: *Drone Delivery Models for Healthcare*. In: *2017 Hawaii International Conference on System Sciences (HICSS)*, 2017.

[133] E. Semsch, M. Jakob, D. Pavlicek und M. Pechoucek: *Autonomous UAV Surveillance in Complex Urban Environments*. In: *2009 IEEE/WIC/ACM International Joint Conference on Web Intelligence and Intelligent Agent Technology*, Bd. 2, S. 82–85, Sep. 2009.

[134] S. Shen, Y. Mulgaonkar, N. Michael und V. Kumar: *Multi-sensor fusion for robust autonomous flight in indoor and outdoor environments with a rotorcraft MAV*. In: *2014 IEEE International Conference on Robotics and Automation (ICRA)*, S. 4974–4981, 2014.

[135] M. Sizintsev, A. Rajvanshi, H. Chiu, K. Kaighn, S. Samarasekera und D. P. Snyder: *Multi-Sensor Fusion for Motion Estimation in Visually-Degraded Environments*. In: *2019 IEEE International Symposium on Safety, Security, and Rescue Robotics (SSRR)*.

[136] Skydio: *Technische Daten der Skydio-2-Plus*. `https://www.skydio.com/skydio-2-plus`.

[137] J. Sola: *Quaternion kinematics for the error-state KF*. 2017.

[138] B. Soni und A. Sowmya: *Victim detection and localisation in an urban disaster site*. In: *2013 IEEE International Conference on Robotics and Biomimetics (ROBIO)*, S. 2142–2147, 2013.

[139] Stanford Artificial Intelligence Laboratory et al.: *Robotic Operating System*. `https://www.ros.org`.

[140] K. Sun, K. Mohta, B. Pfrommer, M. Watterson, S. Liu, Y. Mulgaonkar, C. J. Taylor und V. Kumar: *Robust Stereo Visual Inertial Odometry for Fast Autonomous Flight*. IEEE Robotics and Automation Letters, 3(2):965–972, 2018.

[141] M. Tan, R. Pang und Q. V. Le: *EfficientDet: Scalable and Efficient Object Detection*. In: *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, S. 10778–10787, 2020.

[142] S. Thrun und M. Montemerlo: *The GraphSLAM Algorithm With Applications to Large-Scale Mapping of Urban Structures*. International Journal on Robotics Research, 25:403–430, 2005.

[143] C. Toth, G. Jóźków, Z. Koppanyi und D. Grejner-Brzezinska: *Positioning Slow-Moving Platforms by UWB Technology in GPS-Challenged Areas*. Journal of Surveying Engineering, 143, Nov. 2017.

[144] Uber Elevate: *Fast-Forwarding to a Future of On-Demand Urban Air Transportation*. 2016.

[145] S. Umeyama: *Least-squares estimation of transformation parameters between two point patterns*. IEEE Transactions on Pattern Analysis and Machine Intelligence, 13(4):376–380, 1991.

[146] B. für Digitales und Verkehr: *Vernetzte integrierte UAS-gestützte Datenerfassung und -aufbereitung für die Unterstützung von Behörden und Organisationen mit Sicherheitsaufgaben im Bevölkerungsschutz - VISION*. `https://www.bmvi.de/SharedDocs/DE/Artikel/DG/mfund-projekte/vision.html`.

[147] S. Vidas, R. Lakemond, S. Denman, C. Fookes, S. Sridharan und T. Wark: *A Mask-Based Approach for the Geometric Calibration of Thermal-Infrared Cameras*. IEEE Transactions on Instrumentation and Measurement, 61(6):1625–1635, 2012.

[148] P. Virtanen, R. Gommers, T. E. Oliphant, M. Haberland, T. Reddy, D. Cournapeau, E. Burovski, P. Peterson, W. Weckesser, J. Bright, S. J. van der Walt, M. Brett, J. Wilson, K. J. Millman, N. Mayorov, A. R. J. Nelson, E. Jones, R. Kern, E. Larson, C. J. Carey, İ. Polat, Y. Feng, E. W. Moore, J. VanderPlas, D. Laxalde, J. Perktold, R. Cimrman, I. Henriksen, E. A. Quintero, C. R. Harris, A. M. Archibald, A. H. Ribeiro, F. Pedregosa, P. van Mulbregt und SciPy 1.0 Contributors: *SciPy 1.0: Fundamental Algorithms for Scientific Computing in Python*. Nature Methods, 17:261–272, 2020.

[149] K. Wackwitz, L. Schroth und E. Alvarado: *The Drone Industry Barometer 2021 - Drone Industry Insights*. `https://droneii.com/project/drone-industry-barometer`.

[150] J. Wang und E. Olson: *AprilTag 2: Efficient and robust fiducial detection*. In: *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, October 2016.

[151] J. Wang und E. Olson: *AprilTag 2: Efficient and robust fiducial detection*. In: *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, October 2016.

[152] J. Wendel: *Integrierte Navigationssysteme 2.A.: Sensordatenfusion, GPS und Inertiale Navigation*. Oldenbourg Wissenschaftsverlag, 2011.

[153] Wingcopter: *German Airways and Wingcopter agree to cooperate and drive the use of drones with offshore deliveries*. `https://wingcopter.com/german-airways`.

[154] Wingtra: *WingtrOne - Technical specification*. https://wingtra.b-cdn.net/wp-content/uploads/Wingtra-Technical-Specifications.pdf.

[155] A. Woodworth: *The Dallas Flyer Episode 11: Wing to Launch Drone Delivery Service in Texas April 7th*. https://blog.wing.com/2022/04/the-dallas-flyer-episode-11-wing-to.html.

[156] J. S. Wynn und T. W. McLain: *Visual Servoing for Multirotor Precision Landing in Daylight and After-Dark Conditions*. In: *2019 International Conference on Unmanned Aircraft Systems (ICUAS)*, S. 1242–1248, 2019.

[157] Y. Xiong, H. Liu, S. Gupta, B. Akin, G. Bender, Y. Wang, P. J. Kindermans, M. Tan, V. Singh und B. Chen: *MobileDets: Searching for Object Detection Architectures for Mobile Accelerators*. In: *2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, S. 3824–3833, 2021.

[158] J. Zhang und S. Singh: *LOAM: Lidar Odometry and Mapping in Real-time*. Robotics: Science and Systems Conference, 2014.

[159] Z. Zhang und D. Scaramuzza: *A Tutorial on Quantitative Trajectory Evaluation for Visual(-Inertial) Odometry*. In: *IEEE/RSJ Int. Conf. Intell. Robot. Syst. (IROS)*, 2018.

[160] S. Zhao, P. Wang, H. Zhang, Z. Fang und S. Scherer: *TP-TIO: A Robust Thermal-Inertial Odometry with Deep ThermalPoint*. In: *2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, S. 4505–4512, 2020.

Datenblätter

[161] *Datenblatt ADIS16448*. https://www.analog.com/media/en/technical-documentation/data-sheets/adis16448.pdf.

[162] *Datenblatt LEA-M8T*. https://www.u-blox.com/sites/default/files/NEO-LEA-M8T-FW3_DataSheet_%28UBX-15025193%29.pdf.

[163] Coral: *Benchmark Edge TPU Modelle*. https://coral.ai/models/object-detection/.

[164] Coral: *Datenblatt Edge TPU Stick*. https://coral.ai/static/files/Coral-USB-Accelerator-datasheet.pdf.

[165] Dronecode Project Inc.: *Pixhawk 4*. https://docs.px4.io/v1.9.0/en/flight_controller/pixhawk4.html.

[166] T. Instruments: *Datenblatt 60GHz mmWave Sensor EVMs*. https://www.ti.com/lit/ug/swru546d/swru546d.pdf?ts=1638522970178&ref_url=https%253A%252F%252Fwww.ti.com%252Fproduct%252FIWR6843AOP.

[167] T. Instruments: *mmWave Software Development Kit (SDK) v3.0.5*. https://www.ti.com/tool/download/MMWAVE-SDK/03.05.00.04.

[168] Intel: *Datenblatt Intel NUC i7*. https://ark.intel.com/content/www/de/de/ark/products/130392/intel-nuc-kit-nuc7i7dnke.html.

[169] Intel: *Datenblatt RealSense D435*. https://www.intelrealsense.com/wp-content/uploads/2020/06/Intel-RealSense-D400-Series-Datasheet-June-2020.pdf.

[170] Teledyne-FLIR: *Datenblatt Boson 640*. https://flir.netx.net/file/asset/12673/original/attachment.

[171] UP: *Datenblatt UP Core*. https://up-shop.org/up-core-series.html#additional.

Eigene Veröffentlichungen

[172] C. Doer, J. Atman und G. F. Trommer: *GNSS aided Radar Inertial Odometry for UAS Flights in Challenging Conditions*. In: *2022 IEEE Aerospace Conference (AeroConf)*, 2022.

[173] C. Doer, M. Henzler, H. Messner und G. F. Trommer: *HD Map Generation from Vehicle Fleet Data for Highly Automated Driving on Highways*. In: *2020 IEEE Intelligent Vehicles Symposium (IV)*, S. 2014–2020, 2020.

[174] C. Doer, R. Koenig, G. F. Trommer und E. Stumpf: *Autonomous Precision Takeoff and Landing System for VTOLs in Degraded Visual and GNSS Denied Environments*. In: *Deutsche Gesellschaft für Luft- und Raumfahrt - Lilienthal-Oberth e.V.*, 2020.

[175] C. Doer, R. Koenig, G. F. Trommer und E. Stumpf: *Radar Based Autonomous Precision Takeoff and Landing System for VTOLs in GNSS Denied Environments*. In: *2020 International Conference on Unmanned Aircraft Systems (ICUAS)*, S. 922–931, 2020.

[176] C. Doer, G. Scholz, J. Ruppelt und G. F. Trommer: *Inertial sensor data based motion estimation aided by image processing and differential barometry*. In: *2018 25th Saint Petersburg International Conference on Integrated Navigation Systems (ICINS)*, S. 1–10, 2018.

[177] C. Doer, G. Scholz und G. Trommer: *Indoor laser-based SLAM for micro aerial vehicles*. Gyroscopy and Navigation, 8:181–189, Juli 2017.

[178] C. Doer und G. F. Trommer: *An EKF Based Approach to Radar Inertial Odometry*. In: *2020 IEEE International Conference on Multisensor Fusion and Integration for Intelligent Systems (MFI)*, S. 152–159, 2020.

[179] C. Doer und G. F. Trommer: *Radar Inertial Odometry With Online Calibration*. In: *2020 European Navigation Conference (ENC)*, S. 1–10, 2020.

[180] C. Doer und G. F. Trommer: *Radar Visual Inertial Odometry and Radar Thermal Inertial Odometry: Robust Navigation even in Challenging Visual Conditions*. In: *2021 IEEE/RSJ International Conference on Intelligent Rotots and Sytems (IROS)*, 2021.

[181] C. Doer und G. F. Trommer: *Yaw aided Radar Inertial Odometry uisng Manhattan World Assumptions*. In: *2021 28th Saint Petersburg International Conference on Integrated Navigation Systems (ICINS)*, 2021.

[182] C. Doer und G. F. Trommer: *x-RIO: Radar Inertial Odometry with Multiple Radar Sensors and Yaw Aiding*. Gyroscopy and Navigation, 12:329–339, Feb. 2022.

Veröffentlichungen als Koautor

[183] S. Belgardt, C. Doer, S. Hohmann, P. Karg, S. Rothfuss, F. Siebenrock, W. Stork, O. Terzidis, A. Tittel und T. Zwick: *Entrepreneurship für Ingenieure - Konzeption einer innovativen interdisziplinären Lehrveranstaltung*. Handbuch Qualität in Studium, Lehre und Forschung, 76:67–84, 2021.

[184] J. Rauch, C. Doer und G. F. Trommer: *Object Detection on Thermal Images for Unmanned Aerial Vehicles Using Domain Adaption Through Fine-Tuning*. In: *2021 28th Saint Petersburg International Conference on Integrated Navigation Systems (ICINS)*, S. 1–4, 2021.

[185] B. Varga, C. Doer, G. F. Trommer und S. Hohmann: *Validation of a Limit Ellipsis Controller for Rescue Drones*. In: *16th IEEE International Symposium on Applied Computational Intelligence and Informatics*, 2022.

Betreute Arbeiten

[186] J. C. Climent Pardo: *Indoor Navigation for Unmanned Aerial Vehicles*. Bachelorarbeit, Karlsruher Institut für Technologie, Nov. 2020.

[187] Y. Demir: *Sampling basierte Kollisionsvermeidung für UAVs*. Masterarbeit, Karlsruher Institut für Technologie, Mai 2021.

[188] V. Harr: *Real-Time Human Pose Estimation And Classification On Visual And Thermal Images*. Masterarbeit, Karlsruher Institut für Technologie, Juli 2021.

[189] S. Huang: *Motion Estimation based on Thermal Inertial Odometry*. Masterarbeit, Karlsruher Institut für Technologie, Juni 2020.

[190] F. Maisch: *Implementierung und Test Kameramarker-gestützter Start- und Landemanöver für autonome Flugroboter*. Bachelorarbeit, Karlsruher Institut für Technologie, Feb. 2019.

[191] M. M. Nawaz: *Radar Based Material Classification*. Masterarbeit, Karlsruher Institut für Technologie, Sep. 2020.

[192] J. Rauch: *Deep Learning on Thermal Images*. Bachelorarbeit, Karlsruher Institut für Technologie, Sep. 2020.

[193] Q. Zhang: *Mapless Collision Avoidance for UAVs*. Masterarbeit, Karlsruher Institut für Technologie, Juni 2020.

Multisensorielle Navigation und Kartierung in schwierigen Bedingungen

Unbemannte elektrische Fluggeräte verschiedenster Bauart und Größe finden immer mehr Anwendung in vielfältigen Einsatzbereichen. Aktuelle Fluggeräte benötigen jedoch in der Regel einen Piloten zur Steuerung und Überwachung, was die Einsatzmöglichkeiten stark limitiert. Der Schlüssel zur weiteren Verbreitung ist die Autonomie dieser Systeme. Basis hierfür ist eine robuste Navigation sowie die Kartierung der Umgebung. Dies ist insbesondere in schwierigen Bedingungen durch Einschränkung der Sicht oder des GNSS-Empfangs sehr herausfordernd.

Um trotz dieser schwierigen Bedingungen eine genaue und robuste Navigation und Kartierung zu ermöglichen, werden in dieser Arbeit multisensorielle Fusionsverfahren mithilfe von FMCW-Radarsensoren, Kameras und Inertialsensorik entwickelt. Das entworfene Gesamtsystem wird am Szenario eines Rettungseinsatzes demonstriert. Die entwickelten Verfahren sind auf vielfältige Anwendungen autonomer Fluggeräte übertragbar.

ISBN 978-3-7315-1240-0

9 783731 512400

Gedruckt auf FSC-zertifiziertem Papier